Kerdock Codes Determine Unitary 2-Designs

Trung Can*, Narayanan Rengaswamy*, Robert Calderbank, and Henry D. Pfister

Abstract—The non-linear binary Kerdock codes are known to be Gray images of certain extended cyclic codes of length $N = 2^m$ over $\mathbb{Z}_4$. We show that exponentiating these $\mathbb{Z}_4$-valued codewords by $\imath \triangleq \sqrt{-1}$ produces stabilizer states, that are quantum states obtained using only Clifford unitaries. These states are also the common eigenvectors of commuting Hermitian matrices forming maximal commutative subgroups (MCS) of the Pauli group. We use this *quantum* description to simplify the derivation of the *classical* weight distribution of Kerdock codes. Next, we organize the stabilizer states to form $N + 1$ mutually unbiased bases and prove that automorphisms of the Kerdock code permute their corresponding MCS, thereby forming a subgroup of the Clifford group. When represented as symplectic matrices, this subgroup is isomorphic to the projective special linear group PSL$(2, N)$. We show that this automorphism group acts transitively on the Pauli matrices, which implies that the ensemble is *Pauli mixing* and hence forms a unitary 2-design. The *Kerdock* design described here was originally discovered by Cleve et al. (2016), but the connection to classical codes is new which simplifies its description and translation to circuits significantly. Sampling from the design is straightforward, the translation to circuits uses only Clifford gates, and the process does not require ancillary qubits. Finally, we also develop algorithms for optimizing the synthesis of unitary 2-designs on encoded qubits, i.e., to construct *logical* unitary 2-designs. Software implementations are available at https://github.com/nrenga/symplectic-arxiv18a, which we use to provide empirical gate complexities for up to 16 qubits.

Index Terms—Heisenberg-Weyl group, Pauli group, quantum computing, Clifford group, symplectic geometry, Kerdock codes, Delsarte-Goethals codes, Gray map, stabilizer states, mutually unbiased bases, unitary t-designs

I. Introduction

QUANTUM computers promise enormous computational speedups over the best classical supercomputers in certain problems. It has been established that even constant-depth quantum circuits provide an advantage over classical computation [1]. Today these devices are moving out of the lab and becoming generally programmable [2]. Since quantum computers are noisy, they will most likely employ quantum error-correcting codes (QECCs) to ensure reliability of computation. *Classical* error-correcting codes inspired the discovery of the first QECC by Shor [3], the development of CSS codes by Calderbank, Shor and Steane [4], [5], and the development of stabilizer codes [6], [7] by Calderbank, Rains, Shor and Sloane and by Gottesman. A QECC protects $m - k$ logical qubits by embedding them in a system comprising m physical qubits. In fault-tolerant computation, any desired operation on the $m - k$ logical (protected) qubits must be implemented as a physical operation on the m physical qubits that preserves the code space. The most common QECCs are stabilizer QECCs, that are derived from commutative subgroups of the Heisenberg-Weyl group HW_N (also known as the Pauli group), and the simplest quantum circuits are composed of unitary operators from the Clifford group Cliff$_N$ (see Section III for formal definitions). Note that $N = 2^m$ throughout this paper. The Clifford group normalizes HW_N, so each element of Cliff$_N$ induces an automorphism of HW_N under conjugation. Since HW_N elements can be efficiently represented as binary length-$2m$ vectors, Clifford operators can be efficiently represented as $2m \times 2m$ binary *symplectic* matrices that preserve the group structure of HW_N under the binary mapping. A central theme in this paper is that beyond inspiring the construction of QECCs, interactions between the classical and quantum worlds still prove mutually beneficial for other purposes, especially via the binary representations of HW_N and Cliff$_N$.

Prior to running applications, a quantum computer needs to be assessed for its quality, in terms of the fidelity of the quantum operations executed on the system. The *randomized benchmarking* protocol, introduced by Emerson et al. [8], [9], is a well-established scheme that estimates the fidelity of the noise in the system by *twirling* the underlying channel through a randomized sequence of gates and calculating the fidelity of the resulting depolarizing channel. The depolarizing channel can be interpreted as the quantum analogue of the binary symmetric channel in classical communication theory. Since the fidelity is shown to be an invariant of this twirling process, the protocol indeed estimates the fidelity of the actual noise (under some additional assumptions). This protocol works on the physical operations in the computer, so the fidelity estimates then need to be translated into the quality of the *logical* operations (on the protected qubits) for an error-corrected quantum computer. Since this translation might not be reliable, the procedure has recently been extended to the *logical randomized benchmarking* protocol [10] that directly estimates the fidelity of the logical operations more reliably.

Randomized benchmarking requires that the sequence of gates be sampled from an ensemble of unitaries that form a *unitary 2-design*. A unitary t-design is an ensemble of unitaries, bestowed with a probability distribution, that approximates the Haar distribution on the unitary group up to the t-th moment. Section VI makes this precise by discussing linear maps called *twirls* over the full unitary ensemble and the finite ensemble. When the two linear maps coincide the finite ensemble is a unitary t-design. It is easy to analyze protocols that randomly sample unitary matrices with respect to the Haar measure, but such sampling is infeasible, hence

*These two authors contributed equally to this work.

T. Can is currently with the Department of Mathematics, California Institute of Technology, Pasadena, CA 91125, USA. Most of this work was conducted while he was with the Department of Mathematics, Duke University, Durham, North Carolina 27708, USA. Email: tcan@caltech.edu

N. Rengaswamy, R. Calderbank and H. D. Pfister are with the Department of Electrical and Computer Engineering, Duke University, Durham, North Carolina 27708, USA. Email: {narayanan.rengaswamy, robert.calderbank, henry.pfister}@duke.edu

the interest in finite ensembles of unitary matrices that approximate the Haar distribution. Cliff_N is known to be a unitary 3-design [11], and the proof by Webb involves the concepts of Pauli mixing and Pauli 2-mixing, which we introduce in Section VI. However, Cliff_N has size $2^{m^2+2m}\prod_{j=1}^{m}(4^j-1)$ (up to scalars $e^{2\pi\imath\theta}, \theta \in \mathbb{R}$) which is much larger than the lower bound of $\approx 2^{4m}$ for any unitary 2-design, established by Gross, Audenaert and Eisert [12], and by Roy and Scott [13]. They also discuss the existence of Clifford ensembles that saturate this bound. Although random circuits are known to be exact or approximate unitary 2-designs [14], [15], *deterministic* constructions of such ensembles facilitate practical realizations. Cleve et al. [16] have recently found an explicit subgroup of Cliff_N that is a unitary 2-design. The central contribution of this paper is to use the *classical* Kerdock codes to simplify the construction of this *quantum* unitary 2-design and its translation to circuits (see Section VI).

There are many other applications of unitary 2-designs. The first is quantum data hiding (the LOCC model described in [17]), where the objective is to hide classical information from two parties who each share part of the data but are only allowed to perform local operations and classical communication. The data hiding protocol can be implemented by sampling randomly from the full unitary group but it is sufficient to sample randomly from a unitary 2-design. Other applications of unitary 2-designs are the fidelity estimation of quantum channels [18], quantum state and process tomography [19], and more recently minimax quantum state estimation [20]. In quantum information theory, they have been used extensively in the analysis of decoupling [13], [15], [21], [22].

The next section discusses the flow of ideas in the rest of the paper, and points to related work and applications.

II. Main Ideas and Discussion

The first result in the paper is to use the (quantum) commutative subgroups of HW_N to simplify the derivation of weight distributions for the (classical) Kerdock codes.

Recall that Hermitian matrices that commute can be simultaneously diagonalized. The Kerdock and Delsarte-Goethals binary codes are unions of cosets of the first order Reed-Muller (RM) code $\mathrm{RM}(1,m)$, the cosets are in one-to-one correspondence with symplectic forms, and the weight distribution of a coset is determined by the rank of the symplectic form (see [23, Chapter 15] for more details). Section IV-A reviews the $\mathbb{Z}_4$ description of these codes given in [24]. Section IV-B shows that when we exponentiate codewords in these cosets of $\mathrm{RM}(1,m)$ we obtain a basis of common eigenvectors for a *maximal* commutative subgroup of Hermitian Pauli matrices. The operators that project onto individual lines in a given eigenbasis are invariants of the corresponding subgroup [25], and the distance distribution of the code is determined by the inner product of pairs of such eigenvectors (see [Section IV, Lemma 10]). When calculating weight distributions, this property makes it possible to avoid using Dickson's Theorem [23, Chapter 15] to choose an appropriate representation of the symplectic form. The correspondence between classical and quantum worlds simplifies the calculation (of the weight distribution) given in [23] significantly. In order to demonstrate this simplicity, we provide the proof of the weight distribution here. For most parts of the proof, we only need a brief description of the Kerdock set, Kerdock code, and the Gray map. See Section IV for formal definitions, constructions, and results used in the following proof.

In [26], the *Kerdock set* $P_{\mathrm{K}}(m)$ is defined to be a collection of N binary $m \times m$ symmetric matrices that is characterized by the following properties: it is closed under binary addition, and if $P, Q \in P_{\mathrm{K}}(m)$ are distinct, then $P+Q$ is non-singular. The codewords of the length-N $\mathbb{Z}_4$-linear *Kerdock code* $\mathrm{K}(m)$ can be expressed as $[xPx^T + 2wx^T + \kappa] \pmod 4$, where $x \in \mathbb{F}_2^m$ chooses the symbol index in a codeword, and the codeword is determined by the choice of $w \in \mathbb{F}_2^m, P \in P_{\mathrm{K}}(m)$ and $\kappa \in \mathbb{Z}_4 = \{0,1,2,3\}$. One can obtain complex vectors of quaternary phases $\{1, \imath, -1, -\imath\}$ by raising $\imath = \sqrt{-1}$ to the integer powers in the ($\mathbb{Z}_4$-valued) codeword vector. We will refer to this operation as exponentiating the codeword by $\imath$. The *Gray map* is an isometry from (length-N vectors of quaternary phases, Euclidean metric) to ($\mathbb{Z}_2^{2N}$, Hamming metric) defined by $\mathfrak{g}(1) \triangleq 00, \mathfrak{g}(\imath) \triangleq 01, \mathfrak{g}(-1) \triangleq 11$, and $\mathfrak{g}(-\imath) \triangleq 10$. Note that the domain of the map can equivalently be taken as $\mathbb{Z}_4$ in which case the metric is the Lee metric (defined in Section IV-B). The codewords of the *binary nonlinear* Kerdock code of length 2^{m+1} are obtained as Gray images of the $\mathbb{Z}_4$-valued codewords described above.

Theorem 1: Let m be odd. The weight distribution $A_i, i = 0, \ldots, 2^{m+1}$ of the classical binary Kerdock code of length 2^{m+1} is as follows.

i	0	$2^m - 2^{(m-1)/2}$	2^m	$2^m + 2^{(m-1)/2}$	2^{m+1}
A_i	1	$2^{2m+1} - 2^{m+1}$	$2^{m+2} - 2$	$2^{2m+1} - 2^{m+1}$	1

Proof: We explicitly use Lemma 10 and Corollary 16 to calculate the weight distribution. Fix a vector v of quaternary phases obtained by exponentiating a codeword

$$c_{\mathsf{v}} = [xP_2x^T + 2w_2x^T + \kappa_2]_{x\in\mathbb{F}_2^m}$$

in the $\mathbb{Z}_4$-linear Kerdock code. Consider any vector u of quaternary phases obtained by exponentiating a second codeword

$$c_{\mathsf{u}} = [xP_1x^T + 2w_1x^T + \kappa_1]_{x\in\mathbb{F}_2^m}.$$

(Lemmas 11 and 15 consider $\kappa_1 = \kappa_2 = 0$ so that the eigenvalues are ± 1 (and not in $\{\pm 1, \pm \imath\}$), but we account for these factors here.)

Let n_j be the number of indices $x \in \mathbb{F}_2^m$ for which $(c_{\mathsf{u}})_x - (c_{\mathsf{v}})_x = j$, for $j \in \mathbb{Z}_4$. Since the Gray map preserves the Lee metric, the Hamming distance between the Gray images of c_{u} and c_{v} is $d_H(\mathfrak{g}(c_{\mathsf{u}}), \mathfrak{g}(c_{\mathsf{v}})) = n_1 + 2n_2 + n_3$. Since $n_0 + n_1 + n_2 + n_3 = 2^m$ we simply have to relate n_0 and n_2 to obtain $d_H(\mathfrak{g}(c_{\mathsf{u}}), \mathfrak{g}(c_{\mathsf{v}}))$. Observe that $\langle \mathsf{u}, \mathsf{v} \rangle = (n_0 - n_2) + (n_1 - n_3)\imath$. Lemma 10 implies

$$\begin{aligned} &2^{m+1} - 2\mathrm{Re}[\langle \mathsf{u}, \mathsf{v}\rangle] = 2d_H(\mathfrak{g}(c_{\mathsf{u}}), \mathfrak{g}(c_{\mathsf{v}})) \\ &\Rightarrow d_H(\mathfrak{g}(c_{\mathsf{u}}), \mathfrak{g}(c_{\mathsf{v}})) = 2^m - (n_0 - n_2). \end{aligned} \tag{1}$$

Now we observe three distinct cases for the codeword $c_{\mathsf{u}} - c_{\mathsf{v}}$. Note that there are 2^{2m+2} codewords in $\mathrm{K}(m)$.

(i) $P_1 = P_2, w_1 = w_2$: If $\kappa_1 - \kappa_2 = 0$ then we have the all-zeros codeword, and if $\kappa_1 - \kappa_2 = 2$ then we have the all-ones codeword. However, if $\kappa_1 - \kappa_2 \in \{1, 3\}$ then $n_0 - n_2 = 0$ and this determines two codewords of weight 2^m (more precisely, at distance 2^m from c_{v}).

(ii) $P_1 = P_2, w_1 \neq w_2$: From Corollary 16, irrespective of κ_1, κ_2, we have $\langle \mathsf{u}, \mathsf{v} \rangle = 0$, which implies $n_0 - n_2 = 0$ and hence the distance is 2^m. This determines another $(2^m - 1)2^2 = 2^{m+2} - 4$ codewords of weight 2^m.

(iii) $P_1 \neq P_2$: From Corollary 16 we have $|\langle \mathsf{u}, \mathsf{v} \rangle|^2 = 2^m$, which implies $(n_0 - n_2)^2 + (n_1 - n_3)^2 = 2^m$. Since m is odd, and n_j are non-negative integers, direct calculation shows that this means $(n_0 - n_2)^2 = (n_1 - n_3)^2 = 2^{m-1}$ and therefore $n_0 - n_2 = \pm 2^{(m-1)/2}$. More formally, since the Gaussian integers $\mathbb{Z}[\imath]$ are a unique factorization domain, we have $(n_0 - n_2) + (n_1 - n_3)\imath = (\pm 1 \pm \imath)2^{(m-1)/2}$ and this gives weights $2^m \pm 2^{(m-1)/2}$. Thus, we have $2^{2m+2} - 2^{m+2}$ codewords remaining and it is easy to see that the signs occur equally often. Hence there are $2^{2m+1} - 2^{m+1}$ codewords of each weight. ■

Lemma 11 shows that the (normalized) length-N vectors of quaternary phases obtained by exponentiating Kerdock, or more generally Delsarte-Goethals, codewords are common eigenvectors of maximal commutative subgroups of HW_N. These eigenvectors are called *stabilizer states* in the quantum information literature [27] (also see end of Section III). They can equivalently be described as those quantum states obtainable by applying Clifford unitaries to the basis state $|0\rangle^{\otimes m} = e_0 \otimes \cdots \otimes e_0 = [1, 0, 0, \ldots, 0]^T$ [28]. Clifford elements act by conjugation on HW_N, permuting the maximal commutative subgroups, and fixing the ensemble of stabilizer states (see Section V). The Clifford group is highly symmetric, and it approximates the full unitary group in a way that can be made precise by comparing irreducible representations [25], [29]. Kueng and Gross [30] have shown that the ensemble of stabilizer states is a complex projective 3-design; given a polynomial of degree at most 3, the integral over the N-sphere can be calculated by evaluating the polynomial at stabilizer states, and taking a finite sum. Stabilizer states also find application as measurements in the important classical problem of phase retrieval, where an unknown vector is to be recovered from the magnitudes of the measurements (see Kueng, Zhu and Gross [31]). A third application is unsourced multiple access, where there is a large number of devices (messages) each of which transmits (is transmitted) infrequently. This provides a model for machine-to-machine communication in the Internet-of-Things (IoT), including the special case of radio-frequency identification (RFID), as well as neighbor discovery in ad-hoc wireless networks. Here, Thompson and Calderbank [32] have shown that stabilizer states associated with Delsarte-Goethals codes support a fast algorithm for unsourced multiple access that scales to 2^{100} devices (arbitrary 100-bit messages).

Section V constructs the $N + 1$ eigenbases (of N stabilizer states each) determined by the Kerdock code of length N over $\mathbb{Z}_4$, and shows that the corresponding maximal commutative subgroups partition the non-identity Hermitian Pauli matrices. The eigenbases are mutually unbiased, so that unit vectors u, v in different eigenbases satisfy $|\langle \mathsf{u}, \mathsf{v} \rangle| = N^{-\frac{1}{2}}$, and hence each eigenbasis looks like noise to the other eigenbases. The Kerdock ensemble of $N(N + 1)$ complex lines is extremal; Calderbank et al. [33] have shown that any collection of unit vectors for which pairwise inner products have absolute value 0 or $N^{-\frac{1}{2}}$ has size at most $N^2 + N$, and that any extremal example must be a union of eigenbases. The group of Clifford symmetries of this ensemble, represented as binary symplectic matrices, is shown to be isomorphic to the projective special linear group $\mathrm{PSL}(2, N)$, and hence its size is $(N + 1)N(N - 1) = 2^{3m} - 2^m$. We note that the Kerdock ensemble also appears in the work of Tirkkonen et al. [34].

Section VI defines a graph $\mathbb{H}_N$ whose vertices are labeled by (scalar multiples of) all non-identity (Hermitian) Pauli matrices, where two matrices (vertices) are joined (by an edge) if and only if they commute. This graph is shown to be *strongly regular*; every vertex has the same degree, and the number of vertices joined to two given vertices depends only on whether the two vertices are joined or not joined. The automorphism group of this graph is the group of binary symplectic matrices $\mathrm{Sp}(2m, \mathbb{F}_2)$. A subgroup of Cliff_N containing HW_N is proven to be Pauli mixing if it acts transitively on vertices, and Pauli 2-mixing if it acts transitively on edges and on non-edges. These properties imply that Pauli mixing ensembles are unitary 2-designs and Pauli 2-mixing ensembles are unitary 3-designs [11]. The Clifford symmetries of the Kerdock ensemble (of stabilizer states), again represented as symplectic matrices, are shown to be transitive on the vertices of $\mathbb{H}_N$ and hence a unitary 2-design. Since the Clifford symmetries include all Hermitian Paulis, in addition to $\mathrm{PSL}(2, N)$, the size of the Kerdock unitary 2-design is $N^5 - N^3 \approx 2^{5m}$, which almost saturates the bound by Gross et al. [12] discussed above. The next step is to translate these symmetry elements into circuits for, say, randomized benchmarking.

T. Can has developed an algorithm [35] that factors a $2m \times 2m$ binary symplectic matrix into a product of at most 6 elementary symplectic matrices of the type shown in Table I. The target symplectic matrix maps the (Hadamard) dual basis $X_N = E([I_m \mid 0]), Z_N = E([0 \mid I_m])$ (see Section III for notation) to a dual basis X'_N, Z'_N. Then, row and column operations by the elementary matrices return X'_N, Z'_N to the original pair X_N, Z_N thereby producing a decomposition of the target symplectic matrix.

Section VI uses this decomposition to simplify the translation of the Kerdock unitary 2-design into circuits. The elementary symplectic matrices appearing in the product can be related to the Bruhat decomposition of the symplectic group $\mathrm{Sp}(2m, \mathbb{F}_2)$ (see [36]). When the algorithm is run in reverse it produces a random Clifford matrix that can serve as an approximation to a random unitary matrix. This is an instance of the subgroup algorithm [37] for generating uniform random variables. The algorithm has complexity $O(m^3)$ and uses $O(m^2)$ random bits, which is order optimal given the order of the symplectic group $\mathrm{Sp}(2m, \mathbb{F}_2)$ (cf. [38]). We note that the problem of selecting a unitary matrix uniformly at random finds application in machine learning (see [39] and the references therein). The algorithm developed by Can is similar to that developed by Jones, Osipov and Rokhlin [40] in that it alternates (partial) Hadamard matrices and diagonal matrices; the difference is that the unitary 3-design property of the Clifford group [11] provides randomness guarantees.

Finally, Section VII constructs *logical* unitary 2-designs that can be applied in the logical randomized benchmarking protocol of Combes et al. [10]. In prior work [41], we have developed a mathematical framework for synthesizing *all* physical circuits that implement a logical Clifford operator (on the encoded qubits) for stabilizer codes (up to equivalence classes and ignoring stabilizer freedom). Circuit synthesis is enabled by representing the desired physical Clifford operator as a $2m \times 2m$ binary symplectic matrix. For an $[\![m, m-k]\!]$ stabilizer code, every logical Clifford operator is shown to have $2^{k(k+1)/2}$ symplectic solutions, and these are enumerated efficiently using symplectic transvections, thus enabling optimization with respect to a suitable metric. See https://github.com/nrenga/symplectic-arxiv18a for implementations.

It is now well-known that different codes yield efficient (e.g., low-depth) implementations of different logical operators. However, computing environments can change dynamically so that qubits or qubit links might have varying fidelity, and thus low-depth alone might not be desirable. Under such circumstances it is necessary to leverage all degrees of freedom in implementing a logical operator, and a compiler might use the above framework for this purpose. More generally, a compiler might usefully switch between several codes [42] *dynamically*, depending on the state of the system. Then this algorithm enables the compiler to be able to determine logical operators for a code quickly depending on the user-input circuit (on the *protected* qubits).

Section VII provides a proof of concept implementation of the Kerdock unitary 2-design on the protected (logical) qubits of the $[\![6, 4, 2]\!]$ CSS code using the above logical Clifford synthesis algorithm. The logical randomized benchmarking protocol requires a unitary 2-design on the logical qubits, and Combes et al. use the full Clifford group for this purpose, which is much larger than the Kerdock design as shown above.

In summary, the purpose of this paper is to emphasize that interactions between the classical and quantum domains still prove mutually beneficial, as much as they helped inspire the first QECC more than two decades back. Specifically, we make four main *theoretical* contributions:

1) Use of quantum concepts to simplify the calculation of classical weight distributions of several families of nonlinear binary codes [23], [43]–[49].
2) Elementary description of symmetries of the Kerdock code, and the $N^2 + N$ stabilizer states determined by this code [26], [32], [34], [50].
3) Demonstration that the symmetry group of the Kerdock code is a unitary 2-design and that sampling from it is straightforward. Introduction of elementary methods for translation to circuits without using ancillary qubits.
4) Provide a proof of concept construction for unitary 2-designs on the logical qubits of a stabilizer code [10], [41].

We also provide *software* implementations of all algorithms, at https://github.com/nrenga/symplectic-arxiv18a. Using this utility, we provide empirical estimates of the gate complexity for circuits obtained from the Kerdock design. We believe this paves the way for employing this design in several applications, specifically in randomized benchmarking [9], [51].

III. The Heisenberg-Weyl and Clifford Groups

Quantum error-correcting codes serve to protect qubits involved in quantum computation, and this section summarizes the mathematical framework introduced in [4], [6], [7], [52], and described more completely in [53] and [41]. In this framework for fault-tolerant quantum computation, Clifford operators on the N-dimensional complex space afforded by m qubits are represented as $2m \times 2m$ binary symplectic matrices. This is an exponential reduction in size, and the symplectic matrices serve as a binary control plane for the quantum computer.

Remark 2: Throughout the paper, we adopt the convention that all binary vectors are row vectors, and $\mathbb{Z}_4$-, real- or complex-valued vectors are column vectors, where $\mathbb{Z}_4$ is the ring of integers modulo 4. The values $\imath^{\kappa}$, where $\imath \triangleq \sqrt{-1}, \kappa \in \mathbb{Z}_4$, are called *quaternary phases*.

A single qubit is a 2-dimensional Hilbert space, and a quantum state v is a superposition of the two states $e_0 \triangleq [1, 0]^T, e_1 \triangleq [0, 1]^T$ which form the *computational basis*. Thus $\mathsf{v} = \alpha e_0 + \beta e_1$, where $\alpha, \beta \in \mathbb{C}$ satisfy $|\alpha|^2 + |\beta^2| = 1$ as per the Born rule [54, Chapter 3]. The *Pauli* matrices are

$$I_2,\ X \triangleq \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},\ Z \triangleq \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix},\ Y \triangleq \imath XZ = \begin{bmatrix} 0 & -\imath \\ \imath & 0 \end{bmatrix}, \tag{2}$$

where $\imath \triangleq \sqrt{-1}$ and I_2 is the 2×2 identity matrix [55, Chapter 10]. We may express an arbitrary *pure* quantum state v as

$$\mathsf{v} = (\alpha_0 I_2 + \alpha_1 X + \imath \alpha_2 Z + \alpha_3 Y)\ e_0,\ \text{where } \alpha_i \in \mathbb{R}. \tag{3}$$

We describe m-qubit states by (linear combinations of) m-fold Kronecker products of computational basis states, or equivalently by m-fold Kronecker products of Pauli matrices.

Given row vectors $a, b \in \mathbb{F}_2^m$ define the m-fold Kronecker product

$$D(a, b) \triangleq X^{a_1} Z^{b_1} \otimes \cdots \otimes X^{a_m} Z^{b_m} \in \mathbb{U}_N,\ N \triangleq 2^m, \tag{4}$$

where $\mathbb{U}_N$ denotes the group of all $N \times N$ unitary operators. The *Heisenberg-Weyl group* HW_N (also called the *m-qubit Pauli group*) consists of all operators $\imath^{\kappa} D(a, b)$, where $\kappa \in \mathbb{Z}_4 \triangleq \{0, 1, 2, 3\}$. The order is $|HW_N| = 4N^2$ and the *center* of this group is $\langle \imath I_N \rangle \triangleq \{I_N, \imath I_N, -I_N, -\imath I_N\}$, where I_N is the $N \times N$ identity matrix. Multiplication in HW_N satisfies the identity

$$D(a, b) D(a', b') = (-1)^{a' b^T + b' a^T} D(a', b') D(a, b). \tag{5}$$

The standard *symplectic inner product* in $\mathbb{F}_2^{2m}$ is defined as

$$\langle [a, b], [a', b'] \rangle_{\mathrm{s}} \triangleq a' b^T + b' a^T = [a, b]\ \Omega\ [a', b']^T, \tag{6}$$

where the symplectic form $\Omega \triangleq \begin{bmatrix} 0 & I_m \\ I_m & 0 \end{bmatrix}$ (see [6], [41]). Therefore, two operators $D(a,b)$ and $D(a',b')$ commute if and only if $\langle [a,b],[a',b'] \rangle_{\text{s}} = 0$.

The homomorphism $\gamma \colon HW_N \to \mathbb{F}_2^{2m}$ defined by

$$\gamma(\imath^{\kappa} D(a,b)) \triangleq [a,b] \ \forall\ \kappa \in \mathbb{Z}_4 \tag{7}$$

has kernel $\langle \imath I_N \rangle$ and allows us to represent elements of HW_N (up to multiplication by scalars) as binary vectors.

The *Clifford group* Cliff_N consists of all unitary matrices $g \in \mathbb{C}^{N \times N}$ for which $gD(a,b)g^{\dagger} \in HW_N$ for all $D(a,b) \in HW_N$, where $g^{\dagger}$ is the Hermitian transpose of g [53]. Cliff_N is the *normalizer* of HW_N in the unitary group $\mathbb{U}_N$. The Clifford group contains HW_N and its size is $|\text{Cliff}_N| = 2^{m^2+2m} \prod_{j=1}^{m} (4^j - 1)$ (up to scalars $e^{2\pi \imath \theta}, \theta \in \mathbb{R}$) [6]. We regard operators in Cliff_N as *physical* operators acting on quantum states in $\mathbb{C}^N$, to be implemented by quantum circuits. Every operator $g \in \text{Cliff}_N$ induces an automorphism of HW_N by conjugation. Note that the inner automorphisms induced by matrices in HW_N preserve every conjugacy class $\{\pm D(a,b)\}$ and $\{\pm \imath D(a,b)\}$, because (5) implies that elements in HW_N either commute or anti-commute. Matrices $D(a,b)$ are symmetric or anti-symmetric according as $ab^T = 0$ or 1, hence the matrix

$$E(a,b) \triangleq \imath^{ab^T} D(a,b) \tag{8}$$

is Hermitian. Note that $E(a,b)^2 = I_N$. The automorphism induced by a Clifford element g satisfies

$$gE(a,b)g^{\dagger} = \pm E\left([a,b]F_g\right), \text{ where } F_g = \begin{bmatrix} A_g & B_g \\ C_g & D_g \end{bmatrix} \tag{9}$$

is a $2m \times 2m$ binary matrix that preserves symplectic inner products: $\langle [a,b]F_g, [a',b']F_g \rangle_{\text{s}} = \langle [a,b],[a',b'] \rangle_{\text{s}}$. Hence F_g is called a *binary symplectic matrix* and the symplectic property reduces to $F_g \Omega F_g^T = \Omega$, or equivalently

$$A_g B_g^T = B_g A_g^T,\ C_g D_g^T = D_g C_g^T,\ A_g D_g^T + B_g C_g^T = I_m. \tag{10}$$

(See [56] for an extensive discussion on general symplectic geometry and quantum mechanics.) The symplectic property encodes the fact that the automorphism induced by g must respect commutativity in HW_N. Let $\text{Sp}(2m,\mathbb{F}_2)$ denote the group of symplectic $2m \times 2m$ matrices over $\mathbb{F}_2$. The map $\phi \colon \text{Cliff}_N \to \text{Sp}(2m,\mathbb{F}_2)$ defined by

$$\phi(g) \triangleq F_g \tag{11}$$

is a homomorphism with kernel HW_N, and every Clifford operator projects onto a symplectic matrix F_g. Thus, HW_N is a normal subgroup of Cliff_N and $\text{Cliff}_N / HW_N \cong \text{Sp}(2m,\mathbb{F}_2)$. This implies that the size is $|\text{Sp}(2m,\mathbb{F}_2)| = 2^{m^2} \prod_{j=1}^{m} (4^j - 1)$ (also see [6]). Table I lists elementary symplectic transformations F_g, that generate the binary symplectic group $\text{Sp}(2m,\mathbb{F}_2)$, and the corresponding unitary automorphisms $g \in \text{Cliff}_N$, which together with HW_N generate Cliff_N. (See [41, Appendix I] for a discussion on the Clifford gates and circuits corresponding to these transformations.)

TABLE I

A GENERATING SET OF SYMPLECTIC MATRICES AND THEIR CORRESPONDING UNITARY OPERATORS.

The number of 1s in Q and P directly relates to number of gates involved in the circuit realizing the respective unitary operators (see [41, Appendix I]). The N coordinates are indexed by binary vectors $v \in \mathbb{F}_2^m$, and e_v denotes the standard basis vector in $\mathbb{C}^N$ with an entry 1 in position v and all other entries 0. Here H_{2^t} denotes the Walsh-Hadamard matrix of size 2^t, $U_t = \text{diag}\left(I_t, 0_{m-t}\right)$ and $L_{m-t} = \text{diag}\left(0_t, I_{m-t}\right)$.

Symplectic Matrix F_g	Clifford Operator g
$\Omega = \begin{bmatrix} 0 & I_m \\ I_m & 0 \end{bmatrix}$	$H_N = H_2^{\otimes m}$
$L_Q = \begin{bmatrix} Q & 0 \\ 0 & Q^{-T} \end{bmatrix}$	$\ell_Q : e_v \mapsto e_{vQ}$
$T_P = \begin{bmatrix} I_m & P \\ 0 & I_m \end{bmatrix}; P = P^T$	$t_P \ = \ \text{diag}\left(\imath^{vPv^T \bmod 4}\right)$
$G_t = \begin{bmatrix} L_{m-t} & U_t \\ U_t & L_{m-t} \end{bmatrix}$	$g_t = H_{2^t} \otimes I_{2^{m-t}}$

We use commutative subgroups of HW_N to define resolutions of the identity. A *stabilizer* is a subgroup S of HW_N generated by commuting Hermitian matrices of the form $\pm E(a,b)$, with the additional property that if $E(a,b) \in S$ then $-E(a,b) \notin S$ [55, Chapter 10]. The operators $\frac{I_N \pm E(a,b)}{2}$ project onto the ± 1 eigenspaces of $E(a,b)$, respectively.

Remark 3: Since all elements of S are unitary, Hermitian and commute with each other, they can be diagonalized simultaneously with respect to a common orthonormal basis, and their eigenvalues are ± 1 with algebraic multiplicity $N/2$. We refer to such a basis as the *common eigenbasis* or simply *eigenspace* of the subgroup S, and to the subspace of eigenvectors with eigenvalue $+1$ as the *+1 eigenspace* of S.

If the subgroup S is generated by $E(a_i,b_i), i = 1,\ldots,k$, then the operator

$$\frac{1}{2^k} \prod_{i=1}^{k} (I_N + E(a_i,b_i)) \tag{12}$$

projects onto the 2^{m-k}-dimensional subspace $V(S)$ fixed pointwise by S, i.e., the $+1$ eigenspace of S. The subspace $V(S)$ is the *stabilizer code* determined by S. We use the notation $[\![m, m-k]\!]$ code to represent that $V(S)$ encodes $m-k$ *logical* qubits into m *physical* qubits.

Let $\gamma(S)$ denote the subspace of $\mathbb{F}_2^{2m}$ formed by the binary representations of the elements of S using the homomorphism γ in (7). A generator matrix for $\gamma(S)$ is

$$G_S \triangleq [a_i, b_i]_{i=1,\ldots,k} \in \mathbb{F}_2^{k \times 2m} \text{ s.t. } G_S\ \Omega\ G_S^T = 0, \tag{13}$$

where 0 is the $k \times k$ matrix with all entries zero.

Given a stabilizer S with generators $E(a_i,b_i), i = 1,\ldots,k$, we can define 2^k subgroups $S_{\epsilon_1 \cdots \epsilon_k}$ where the index $(\epsilon_1 \cdots \epsilon_k)$ represents that $S_{\epsilon_1 \cdots \epsilon_k}$ is generated by $\epsilon_i E(a_i,b_i)$, for $\epsilon_i \in \{\pm 1\}$. Note that the operator

$$\Pi_{\epsilon_1 \cdots \epsilon_k} \triangleq \frac{1}{2^k} \prod_{i=1}^{k} (I_N + \epsilon_i E(a_i,b_i)) \tag{14}$$

projects onto $V(S_{\epsilon_1 \cdots \epsilon_k})$, and that

$$\sum_{(\epsilon_1,\ldots,\epsilon_k) \in \{\pm 1\}^k} \Pi_{\epsilon_1 \cdots \epsilon_k} = I_N. \tag{15}$$

Hence the subspaces $V(S_{\epsilon_1\cdots\epsilon_k})$, or equivalently the subgroups $S_{\epsilon_1\cdots\epsilon_k}$, provide a resolution of the identity, and elements (errors) in HW_N simply permute these subspaces (under conjugation).

Given an $[\![m, m-k]\!]$ stabilizer code, it is possible to perform encoded quantum computation in any of the subspaces $V(S_{\epsilon_1\cdots\epsilon_k})$ by synthesizing appropriate logical Clifford operators (see [41] for algorithms). If we think of these subspaces as *threads*, then a computation starts in one thread and jumps to another when an error (from HW_N) occurs. Quantum error-correcting codes enable error control by identifying the jump that the computation has made. Identification makes it possible to modify the computation in flight instead of returning to the initial subspace and restarting the computation. The idea of tracing these threads is called as *Pauli frame tracking* in the literature (see [57] and references therein).

A stabilizer group S defined by $k = m$ generators is called a *maximal commutative subgroup* of HW_N and $\gamma(S)$ is called a *maximal isotropic subspace* of $\mathbb{F}_2^{2m}$. The generator matrix G_S has rank m and can be row-reduced to $[0 \mid I_m]$ if $S = Z_N \triangleq \{E(0,b) \colon b \in \mathbb{F}_2^m\}$, or to the form $[I_m \mid P]$ if S is disjoint from Z_N. We will denote these subgroups as $E([0 \mid I_m])$ and $E([I_m \mid P])$, respectively. The condition $G_S \Omega G_S^T = 0$ implies $P = P^T$, and any element of $\gamma(S)$ can be expressed in the form $[a, aP]$ for some $a \in \mathbb{F}_2^m$. Note that $E([I_m \mid 0]) = X_N \triangleq \{E(a,0) \colon a \in \mathbb{F}_2^m\}$. Since $\dim V(S) = 2^{m-m} = 1$, the subgroup S fixes exactly one vector. The N eigenvectors in an orthonormal eigenbasis for S are defined up to an overall phase and called *stabilizer states* [27], [28]. The number of non-zero entries in a stabilizer state is determined by the intersection of S with Z_N [58].

IV. Weight Distributions of Kerdock Codes

Kerdock codes were first constructed as non-linear binary codes [44], as was the Goethals code [46] and the Delsarte-Goethals codes [48]. In this section, we describe the Kerdock and Delsarte-Goethals codes as linear codes over $\mathbb{Z}_4$, the ring of integers modulo 4. These $\mathbb{Z}_4$-linear codes were constructed by Hammons et al. [24] as Hensel lifts of binary cyclic codes, and this description requires Galois rings. The description given in Section IV-A requires finite field arithmetic, but is entirely binary and follows [26]. Our construction of unitary 2-designs in Section VI uses the matrices that are defined in Section IV-A. In Section IV-B, we make a connection between the Kerdock and Delsarte-Goethals codes and maximal commutative subgroups of HW_N via stabilizer states, use this relation to compute inner products between stabilizer states, and hence calculate weight distributions of Kerdock codes.

A. Kerdock and Delsarte-Goethals Sets [26]

The finite field $\mathbb{F}_{2^m}$ is obtained from the binary field $\mathbb{F}_2$ by adjoining a root α of a primitive irreducible polynomial $p(x)$ of degree m [59]. The elements of $\mathbb{F}_{2^m}$ are polynomials in α of degree at most $m-1$, with coefficients in $\mathbb{F}_2$, and we will identify the polynomial $z_0 + z_1\alpha + \ldots + z_{m-1}\alpha^{m-1}$ with the binary (row) vector $[z_0, z_1, \ldots, z_{m-1}]$.

The *Frobenius map* $f \colon \mathbb{F}_{2^m} \to \mathbb{F}_{2^m}$ is defined by $f(x) \triangleq x^2$, and the *trace map* $\text{Tr} \colon \mathbb{F}_{2^m} \to \mathbb{F}_2$ is defined by

$$\text{Tr}(x) \triangleq x + x^2 + \ldots + x^{2^{m-1}}. \tag{16}$$

Since $(x+y)^2 = x^2 + y^2$ for all $x, y \in \mathbb{F}_{2^m}$, the trace is linear over $\mathbb{F}_2$. The trace inner product $\langle x, y\rangle_{\text{tr}} = \text{Tr}(xy)$ defines a symmetric bilinear form, so there exists a binary symmetric matrix W for which $\text{Tr}(xy) = xWy^T$. In fact

$$W_{ij} = \text{Tr}\left(\alpha^i\alpha^j\right), \; i, j = 0, 1, \ldots, m-1. \tag{17}$$

The matrix W is non-singular since the trace inner product is non-degenerate (if $\text{Tr}(xz) = 0$ for all $z \in \mathbb{F}_{2^m}$ then $x = 0$). Observe that W is a Hankel matrix, since if $i + j = h + k$ then $\text{Tr}(\alpha^i\alpha^j) = \text{Tr}(\alpha^h\alpha^k)$. The matrix W can be interpreted as the primal-to-dual-basis conversion matrix for $\mathbb{F}_{2^m}$, with the primal basis being $\{1, \alpha, \alpha^2, \ldots, \alpha^{m-1}\}$ (see [16]).

The Frobenius map $f(x) = x^2$ is linear over $\mathbb{F}_2$, so there exists a binary matrix R for which $f(x) \equiv xR$. Since

$$\begin{aligned} f(x_0 + x_1\alpha + \ldots + x_{m-1}\alpha^{m-1}) \\ = x_0 + x_1\alpha^2 + \ldots + x_{m-1}\alpha^{2(m-1)}, \end{aligned} \tag{18}$$

the rows of R are the vectors representing the field elements $\alpha^{2i}, i = 0, \ldots, m-1$. Note that square roots exist for all elements of $\mathbb{F}_{2^m}$ since R is invertible.

We write multiplication by $z \in \mathbb{F}_{2^m}$ as a linear transformation $xz \equiv xA_z$. For $z = 0, A_0 = 0$, and for $z = \alpha^i$ the matrix $A_z = A^i$ for $i = 0, 1, \ldots, 2^m - 2$, where A is the matrix that represents multiplication by the primitive element α. The matrix A is the *companion matrix* of the primitive irreducible polynomial $p(x) = p_0 + p_1x + \ldots + p_{m-1}x^{m-1} + x^m$ over the binary field. Thus

$$A \triangleq \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ p_0 & p_1 & p_2 & \cdots & p_{m-1} \end{bmatrix}, \tag{19}$$

and we have chosen A rather than A^T as the companion matrix since we are representing field elements in $\mathbb{F}_{2^m}$ by row vectors (rather than column vectors).

Lemma 4: The matrices A_z, W, and R^i, for $i \in [m]$, satisfy:

(a) $A_zA_x = A_xA_z = A_{xz}$;
(b) $A_x + A_z = A_{x+z}$;
(c) $A_zW = WA_z^T$;
(d) $R^iA_x^{2^i} = A_xR^i$, $R^iA_x^2 = A_x^{2^{1-i}}R^i$, and $R^{-i}A_x^{-2} = A_x^{-2^{1+i}}R^{-i}$;
(e) $R^iW = W(R^{-i})^T$ and $W^{-1}R^{-i}W = (R^i)^T$.

Proof: Identities (a) through (d) follow directly from the arithmetic of $\mathbb{F}_{2^m}$. Specifically, for (c), observe that

$$(xA_z)Wy^T = \text{Tr}((xz)y) = \text{Tr}(x(yz)) = xW(yA_z)^T,$$

and (d) can be proven similarly. To prove part (e) we observe $\text{Tr}(x) = \text{Tr}(x^2)$ and verify that for all $x, y \in \mathbb{F}_{2^m}$,

$$(xR^i)Wy^T = \text{Tr}(x^{2^i}y) = \text{Tr}(xy^{2^{-i}}) = xW(R^{-i})^Ty^T. \quad \blacksquare$$

Definition 5: For $0 \leq r \leq (m-1)/2$ and for $\underline{z} = (z_0, z_1, \ldots, z_r) \in \mathbb{F}_{2^m}^{r+1}$ define the bilinear form $\beta_{\underline{z},r}(x,y) \triangleq \text{Tr}[z_0 xy + z_1(x^2y + xy^2) + \ldots + z_r(x^{2^r}y + xy^{2^r})]$. Note that $\beta_{\underline{z},r}(x,y)$ is represented by the binary symmetric matrix

$$P_{\underline{z},r} \triangleq A_{z_0}W + \sum_{i=1}^{r}[A_{z_i}W(R^i)^T + R^iWA_{z_i}^T]. \tag{20}$$

The *Delsarte-Goethals set* $P_{\text{DG}}(m,r)$ consists of all such matrices $P_{\underline{z},r}$. The *Kerdock set* $P_{\text{K}}(m) \triangleq P_{\text{DG}}(m,0)$ consists of all matrices $P_z \triangleq P_{\underline{z},0}$, where $\underline{z} = (z), z \in \mathbb{F}_{2^m}$.

Lemma 6: The Delsarte-Goethals set $P_{\text{DG}}(m,r)$ is an $m(r+1)$-dimensional vector space of symmetric matrices. If $z \neq 0$ then $\text{rank}(P_{\underline{z},r}) \geq m-2r$. Matrices in the Kerdock set $P_{\text{K}}(m)$ are non-singular.

Proof: Closure under addition follows from part (b) of Lemma 4. Observe $\text{Tr}(x) = \text{Tr}(x^2) = \cdots = \text{Tr}(x^{1/2})$. If x is in the nullspace of $P_{\underline{z},r}$, i.e., using its vector representation $xP_{\underline{z},r} = 0$, then $\beta_{\underline{z},r}(x,y) = 0$ for all $y \in \mathbb{F}_{2^m}$ and we obtain

$$\begin{aligned} 0 &= \text{Tr}[z_0xy + z_1(x^2y + xy^2) + \ldots + z_r(x^{2^r}y + xy^{2^r})] \\ &= \text{Tr}[(z_0x)^{2^r}y^{2^r} + (z_1^{2^r}x^{2^{r+1}}y^{2^r} + (z_1x)^{2^{r-1}}y^{2^r}) + \ldots \\ &\qquad \ldots + (z_r^{2^r}x^{2^{2r}}y^{2^r} + (z_rx)y^{2^r})] \\ &= \text{Tr}[y^{2^r}((z_0x)^{2^r} + (z_1^{2^r}x^{2^{r+1}} + (z_1x)^{2^{r-1}}) + \ldots \\ &\qquad \ldots + (z_r^{2^r}x^{2^{2r}} + z_rx))]. \end{aligned}$$

This holds for all y, so $(z_0x)^{2^r} + (z_1^{2^r}x^{2^{r+1}} + (z_1x)^{2^{r-1}}) + \ldots + (z_r^{2^r}x^{2^{2r}} + z_rx)$ must be identically 0, i.e., x is a root of the polynomial. Since the polynomial has at most 2^{2r} roots, the nullspace of $P_{\underline{z},r}$ has dimension at most $2r$, which implies that $\text{rank}(P_{\underline{z},r}) \geq m-2r$. ■

Remark 7: Note that since the dimension of the vector space of all binary $m \times m$ symmetric matrices is $m(m+1)/2$, the set $P_{\text{DG}}(m,(m-1)/2)$ contains *all* possible symmetric matrices. For the remainder of this paper we represent a general symmetric matrix as simply P, thereby dropping the subscripts $\underline{z}, r$ unless necessary. We will continue to represent Kerdock matrices as P_z.

B. Delsarte-Goethals Codes and Weight Distributions

Hammons et al. [24] showed that the classical nonlinear Kerdock and Delsarte-Goethals codes, defined by quadratic forms in [44], [47], are images of linear codes over $\mathbb{Z}_4$ under the Gray map. In this section, we begin by reviewing this construction using the Kerdock and Delsarte-Goethals sets of matrices, and demonstrate that exponentiating these $\mathbb{Z}_4$-valued codewords entry-wise by $\imath$ produces stabilizer states. For stabilizer states of $E([I_m \mid P_{\underline{z}_1,r}])$ and $E([I_m \mid P_{\underline{z}_2,r}])$, we calculate their Hermitian inner products using the trace of certain projection operators, and show that the distribution of inner products depends only on $\text{rank}(P_{\underline{z}_1,r} + P_{\underline{z}_2,r})$. Then, since $\text{rank}(P_z) \in \{0, m\}$ for all $P_z \in P_{\text{K}}(m)$, we compute the weight distribution of Kerdock codes by relating these Hermitian inner products to the histogram of values in the difference between two $\mathbb{Z}_4$-valued codewords. In order to calculate the weight distribution of Delsarte-Goethals codes, we would need to determine the distribution of ranks in the Delsarte-Goethals sets $P_{\text{DG}}(m,r)$. While this question is straightforward for $P_{\text{K}}(m)$, it remains open for general $P_{\text{DG}}(m,r)$ and will be investigated in future work.

Definition 8: The $\mathbb{Z}_4$-linear *Delsarte-Goethals code* is given by

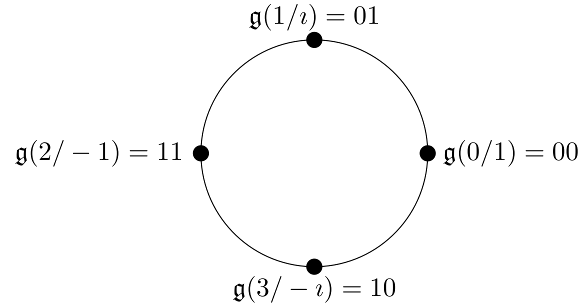
$$\begin{aligned} \text{DG}(m,r) \triangleq \{[xPx^T + 2wx^T + \kappa]_{x\in\mathbb{F}_2^m} &: P \in P_{\text{DG}}(m,r), \\ &w \in \mathbb{F}_2^m, \kappa \in \mathbb{Z}_4\}. \end{aligned} \tag{21}$$

This code has size $2^{m(r+1)+m+2}$ and is of *type* $4^{m+1}2^{mr}$ [49, Section 12.1]. The *Kerdock code* $\text{K}(m) \triangleq \text{DG}(m,0)$.

Here the notation $[xPx^T + 2wx^T + \kappa]_{x\in\mathbb{F}_2^m}$ represents a $\mathbb{Z}_4$-valued column vector with each entry $xPx^T + 2wx^T + \kappa$ (mod 4) indexed by the vector $x \in \mathbb{F}_2^m$.

Definition 9: For $u, v \in \mathbb{Z}_4^N$ the *Lee weight* of u is defined as $w_L(u) \triangleq n_1(u) + 2n_2(u) + n_3(u)$, where $n_\kappa(u)$ denotes the number of entries of u with value κ, and the *Lee distance* (between u and v) is defined as $d_L(u,v) \triangleq w_L(u-v)$.

Figure 1 defines the *Gray* map which assigns integers modulo 4 (or quaternary phases) to binary pairs (see Remark 2 for details). For a vector, the map is applied to each entry and concatenated row-wise to return a row vector, thereby adhering to our convention for binary vectors (see Remark 2). The shortest distance around the circle defines the Lee metric on $\mathbb{Z}_4^N$ and Gray encoding is an isometry from ($\mathbb{Z}_4^N$, Lee metric) to ($\mathbb{Z}_2^{2N}$, Hamming metric). However, since $\mathfrak{g}(1+3) \neq \mathfrak{g}(1) + \mathfrak{g}(3)$, the Gray map is non-linear. Hence the *binary* Kerdock and Delsarte-Goethals codes obtained by Gray-mapping the codewords in $\text{K}(m)$ and $\text{DG}(m,r)$, respectively, are *non-linear* (see [49, Chapter 12]).



$\mathfrak{g}(1/\imath) = 01$
$\mathfrak{g}(2/-1) = 11$
$\mathfrak{g}(0/1) = 00$
$\mathfrak{g}(3/-\imath) = 10$

Fig. 1. The Gray map assigning integers modulo 4 or quaternary phases to binary pairs (that are length-2 row vectors).

The Gray map is also a scaled isometry from (length-N vectors of quaternary phases, squared Euclidean metric) to ($\mathbb{Z}_2^{2N}$, Hamming metric). Note that for this set of quaternary phases, squared Euclidean distance is indeed a metric. This is formalized in the following lemma.

Lemma 10: Let $\mathsf{u}, \mathsf{v} \in \mathbb{C}^N$ be two length-N vectors of quaternary phases. Then

$$\langle \mathsf{u} - \mathsf{v}, \mathsf{u} - \mathsf{v} \rangle = 2d_H(\mathfrak{g}(\mathsf{u}), \mathfrak{g}(\mathsf{v})), \tag{22}$$

where d_H denotes the Hamming distance.

Next, we prove a lemma establishing the relation between $\text{DG}(m,r)$ and the common eigenspace of $E([I_m \mid P])$ determined by a binary symmetric matrix P (see Remark 7). Note that we denote the maximal commutative subgroup determined by the rows of $[I_m \mid P]$ as $E([I_m \mid P])$, and that we do not normalize eigenvectors (stabilizer states) in this section, since the Gray map needs to be applied to quaternary phases.

Lemma 11: Given a binary symmetric matrix P, the (column) vectors $[\imath^{xPx^T+2wx^T}]_{x\in\mathbb{F}_2^m}$ are common eigenvectors of the maximal commutative subgroup $E([I_m \mid P])$. Each eigenvector has Euclidean length $\sqrt{N} = 2^{m/2}$.

Proof: It is possible to prove this result by direct calculation, i.e., by calculating $E(a, aP) \cdot [\imath^{xPx^T+2wx^T}]_{x\in\mathbb{F}_2^m}$ for some $a \in \mathbb{F}_2^m$, but the following argument uses the mathematical framework described in Section III. Note that

$$[\imath^{xPx^T+2wx^T}]_{x\in\mathbb{F}_2^m} = \sum_{x\in\mathbb{F}_2^m} \imath^{xPx^T+2wx^T} e_x, \qquad (23)$$

where e_x denotes the standard basis vector in $\mathbb{C}^N$ with a 1 in the position x and 0 elsewhere.

The columns $[\imath^{2wx^T}]_{x\in\mathbb{F}_2^m} = [(-1)^{wx^T}]_{x\in\mathbb{F}_2^m}$ of the Walsh-Hadamard matrix H_N, where $w \in \mathbb{F}_2^m$ indexes the column, are common eigenvectors of the maximal commutative subgroup $X_N = E([I_m \mid 0])$. The Clifford operator $t_P = \text{diag}\left(\imath^{xPx^T}\right)$ corresponds to the binary symplectic matrix $T_P = \begin{bmatrix} I_m & P \\ 0 & I_m \end{bmatrix}$ (see Table I). Hence conjugation by t_P maps $E([I_m \mid 0])$ to $E([I_m \mid P])$, i.e., $t_P E(a, 0) t_P^\dagger = E(a, aP)$, and so the common eigenvectors of $E([I_m \mid P])$ are

$$t_P \cdot [\imath^{2wx^T}]_{x\in\mathbb{F}_2^m} = [\imath^{xPx^T+2wx^T}]_{x\in\mathbb{F}_2^m}.$$

It is easily verified that for any $a \in \mathbb{F}_2^m$,

$$\left(t_P E(a, 0) t_P^\dagger\right)\ t_P \cdot [\imath^{2wx^T}]_{x\in\mathbb{F}_2^m} = \pm[\imath^{xPx^T+2wx^T}]_{x\in\mathbb{F}_2^m}.$$

■

Now we have the following important observation. Given $\mathsf{v} \in \mathbb{C}^N$, define

$$S_{\mathsf{v}} \triangleq \{g \in HW_N \colon g\mathsf{v} = \alpha\mathsf{v},\ \text{where } \alpha \in \mathbb{C} \text{ and } |\alpha| = 1\}. \qquad (24)$$

Lemma 12:

(a) S_{v} is commutative.

(b) There is a 1-N correspondence between maximal commutative subgroups of HW_N and stabilizer states.

Proof:

(a) If $E(a, b), E(a', b') \in S_{\mathsf{v}}$ then

$$E(a, b)E(a', b')\mathsf{v} = \alpha\alpha'\mathsf{v} = \alpha'\alpha\mathsf{v} = E(a', b')E(a, b)\mathsf{v}.$$

(b) If v is a stabilizer state then S_{v} is a maximal commutative subgroup of HW_N. ■

Remark 13: Any stabilizer state of a maximal commutative subgroup of HW_N *disjoint* from Z_N can be obtained by exponentiating Delsarte-Goethals codewords, and multiplying the quaternary phase vector by $N^{-\frac{1}{2}}$. The maximal commutative subgroups $E([I_m \mid P_z])$ determined by the Kerdock matrices P_z intersect trivially. Together with $Z_N = E([0 \mid I_m])$, they partition all $(N^2 - 1)$ non-identity Hermitian Pauli matrices. Hence, given a non-identity Hermitian Pauli matrix $E(a, b)$, it follows that there is a sign $\epsilon \in \{\pm 1\}$ such that $\epsilon E(a, b)$ is in one of the $N + 1$ subgroups determined by all $P_z \in P_{\mathsf{K}}(m)$ and Z_N.

Therefore, stabilizer states connect the *classical* world of Kerdock and Delsarte-Goethals codes and the *quantum* world of maximal commutative subgroups of HW_N. This synergy has proven successful in several applications [26], [30]–[32], [34], [50] and our construction of a unitary 2-design here, from stabilizer states, is yet another instance.

Remark 14: Note that in Lemma 11 we only considered $\kappa = 0$ while exponentiating codewords $c_{\mathsf{u}} \in \text{DG}(m, r)$. This is to ensure that the resulting eigenvector corresponds to a ± 1 eigenvalue (and not a value in $\{\pm 1, \pm\imath\}$). However, Theorem 1 considers all $\kappa \in \mathbb{Z}_4$ while calculating the weight distribution.

Given $P \in P_{\text{DG}}(m, r)$, we scale the common eigenvectors of $E([I_m \mid P])$ by $\sqrt{N}$ to obtain a set $V(P)$ of length-N vectors of quaternary phases. (Note the similarity to the notation $V(S)$ used in Section III, and observe that here we consider all eigenvectors, albeit unnormalized, and not just the $+1$ eigenspace.) Therefore, if we can compute the Hermitian inner products between these unnormalized stabilizer states then we can use Lemma 10 to calculate the weight distribution of Kerdock and Delsarte-Goethals codes. Note that despite being non-linear codes the weight and distance distributions of these codes coincide, as shown in [23, Chapter 15] (which follows from $\mathbb{Z}_4$-linearity and Gray isometry).

Lemma 15: Let $P_1, P_2 \in P_{\text{DG}}(m, r)$ be distinct. Fix $\mathsf{v} \in V(P_2)$ and let u run through $V(P_1)$. If $\text{rank}(P_1 + P_2) = k$ then

$$|\langle \mathsf{u}, \mathsf{v} \rangle|^2 = \begin{cases} 2^{2m-k} & \text{for } 2^k \text{ eigenvectors } \mathsf{u}, \\ 0 & \text{for } 2^m - 2^k \text{ eigenvectors } \mathsf{u}. \end{cases} \qquad (25)$$

Proof: Let $Q = [I_m \mid P_1] \cap [I_m \mid P_2]$ represent a basis for the subspace formed by intersecting the spaces generated by $[I_m \mid P_1]$ and $[I_m \mid P_2]$. Then $\dim(Q) = m - k$. Let $[a_1, b_1], \ldots, [a_{m-k}, b_{m-k}]$ be a basis for Q, and complete to bases for P_1 and P_2 by adding vectors $[c_j, d_j], j = m - k + 1, \ldots, m$ and $[c'_j, d'_j], j = m-k+1, \ldots, m$ respectively. Since v is fixed, using (14), there are *fixed* $f_i, t_j \in \{\pm 1\}$ such that

$$\left(\frac{1}{\sqrt{N}}\mathsf{v}\right)\left(\frac{1}{\sqrt{N}}\mathsf{v}^\dagger\right) = \prod_{i=1}^{m-k} \frac{(I_N + f_i E(a_i, b_i))}{2} \prod_{j=m-k+1}^{m} \frac{(I_N + t_j E(c'_j, d'_j))}{2},$$

and since the only constraint for u is to be from $V(P_1)$,

$$\left(\frac{1}{\sqrt{N}}\mathsf{u}\right)\left(\frac{1}{\sqrt{N}}\mathsf{u}^\dagger\right) = \prod_{i=1}^{m-k} \frac{(I_N + e_i E(a_i, b_i))}{2} \prod_{j=m-k+1}^{m} \frac{(I_N + s_j E(c_j, d_j))}{2},$$

where $e_i, s_i \in \{\pm 1\}$ are variable. Since $|\langle \mathsf{u}, \mathsf{v} \rangle|^2 = \text{Tr}(\mathsf{u}\mathsf{u}^\dagger\mathsf{v}\mathsf{v}^\dagger)$ it only remains to calculate $\text{Tr}(\mathsf{u}\mathsf{u}^\dagger\mathsf{v}\mathsf{v}^\dagger)$. If $e_i = f_i\ \forall\ i$, then

$$(I_N + e_i E(a_i, b_i))(I_N + f_i E(a_i, b_i)) = 2(I_N + e_i E(a_i, b_i))$$

so that

$$\text{Tr}(\mathsf{u}\mathsf{u}^\dagger\mathsf{v}\mathsf{v}^\dagger) = 2^{m-k}\text{Tr}\left(\prod_{i=1}^{m-k} (I_N + e_i E(a_i, b_i))\right.$$

$$\times \prod_{j=m-k+1}^{m} (I_N + s_j E(c_j, d_j))$$
$$\times \prod_{j=m-k+1}^{m} (I_N + t_j E(c'_j, d'_j))\Big).$$

Expanding the right hand side, the only term with nonzero trace is the identity with trace 2^m. Hence in this case $\mathrm{Tr}(\mathsf{u}\mathsf{u}^\dagger \mathsf{v}\mathsf{v}^\dagger) = 2^{2m-k}$. The k eigenvalues s_j can be freely chosen, so there are 2^k eigenvectors in this case.

If $e_i \neq f_i$ for some i, then

$$(I_N + e_i E(a_i, b_i))(I_N + f_i E(a_i, b_i)) = 0$$

and $\mathrm{Tr}(\mathsf{u}\mathsf{u}^\dagger \mathsf{v}\mathsf{v}^\dagger) = 0$. There are $2^m - 2^k$ such eigenvectors.

Finally, if $k = m$ then $\mathrm{Tr}(\mathsf{u}\mathsf{u}^\dagger \mathsf{v}\mathsf{v}^\dagger) = \mathrm{Tr}(I_N) = 2^m \ \forall \ \mathsf{u}$. ■

Corollary 16: For $P_1, P_2 \in P_{\mathrm{K}}(m)$, since $\mathrm{rank}(P_1 + P_2) \in \{0, m\}$ the inner products are

$$|\langle \mathsf{u}, \mathsf{v}\rangle|^2 = \begin{cases} 0 & \text{if } P_1 = P_2 \text{ and } \mathsf{u} \neq \mathsf{v} \\ 2^m & \text{if } P_1 \neq P_2, \\ 2^{2m} & \text{if } (P_1 = P_2 \text{ and) } \mathsf{u} = \mathsf{v}. \end{cases} \tag{26}$$

This result is the primary tool that allowed us to simplify the derivation of the weight distribution of Kerdock codes in Theorem 1. Note that Theorem 1 is the only result in this section that is restricted to Kerdock codes but not general Delsarte-Goethals codes (and requires m to be odd).

V. Mutually Unbiased Bases from $P_{\mathrm{K}}(m)$

In this section, we will organize the columns of I_N (i.e., the common eigenbasis of Z_N) and all stabilizer states determined by $P_z \in P_{\mathrm{K}}(m)$ into a matrix to form mutually unbiased bases, and analyze its symmetries. This symmetry group will eventually lead to the construction of the unitary 2-design. We first state a result that holds for stabilizer states determined by matrices from general Delsarte-Goethals sets.

Definition 17: Given a collection M of unit vectors in $\mathbb{C}^N$ (Grassmannian lines) the *chordal distance* $\mathrm{chor}(S)$ is given by

$$\mathrm{chor}(M) \triangleq \min_{(\mathsf{u},\mathsf{v}) \in M} \sqrt{1 - |\langle \mathsf{u}, \mathsf{v}\rangle|^2}. \tag{27}$$

It follows from Lemma 15 that the Delsarte-Goethals set $P_{\mathrm{DG}}(m, r)$ determines $2^{m(r+2)}$ complex lines (stabilizer states) in $\mathbb{C}^N$ with chordal distance $\sqrt{1 - 2^{-(m-2r)}}$ (cf. [60]).

Definition 18: Two $N \times N$ unitary matrices U and V are said to be *mutually unbiased* if $|\langle \mathsf{u}, \mathsf{v}\rangle| = N^{-\frac{1}{2}}$ for all columns u of U, and all columns v of V. Each matrix is interpreted as an orthonormal basis and collections of such unitary matrices that are pairwise mutually unbiased are called *mutually unbiased bases* (MUBs). Vectors in each orthonormal basis look like noise to the other bases (due to the small inner product).

Corollary 16, when applied to normalized eigenvectors, shows that the N eigenbases determined by the Kerdock set $P_{\mathrm{K}}(m)$ are mutually unbiased (also see Remark 13). Let $\mathcal{B}_{\mathrm{K}}(m)$ denote the collection of these N eigenbases (of $E([I_m \mid P])$ for all $P \in P_{\mathrm{K}}(m)$) along with the eigenbasis I_N of $E([0 \mid I_m])$. This is a set of $N + 1$ mutually unbiased bases and they determine an ensemble of $N(N + 1)$ complex lines (stabilizer states) that is extremal [33]. In this section, we provide an elementary description of their group of Clifford symmetries.

A. The Kerdock MUBs

Recollect from Section IV-A that the Kerdock matrices are $P_z = A_z W$, where W is a symmetric Hankel matrix with binary entries that satisfies $\mathrm{Tr}(xy) = xWy^T$ and A_z represents multiplication by z, both in $\mathbb{F}_{2^m}$. Using the result of Lemma 11, define N mutually unbiased bases

$$M_z \triangleq t_{P_z} H_N = \mathrm{diag}\left(\imath^{x P_z x^T}\right) H_N, \ z \in \mathbb{F}_{2^m}, \tag{28}$$

where $[H_N]_{x,y} \triangleq \frac{1}{\sqrt{N}} (-1)^{xy^T}$, for $x, y \in \mathbb{F}_2^m$, is the Walsh-Hadamard matrix of order N. Note that $M_0 = H_N$ and that all columns of M_z have Euclidean length $\frac{1}{\sqrt{N}}$. Complete the MUBs by appending the matrix $M_\infty \triangleq I_N$. The common eigenspaces of the maximal commutative subgroup $E([I_m \mid P_z])$ are the columns of M_z and the common eigenspaces of $E([0 \mid I_m])$ are the standard unit coordinate vectors. Hence the set of *Kerdock MUBs*

$$\mathcal{B}_{\mathrm{K}}(m) \triangleq \{I_N, M_z \colon z \in \mathbb{F}_{2^m}\} \tag{29}$$

is a maximal collection of mutually unbiased bases [33].

B. Symmetries of Kerdock MUBs

Let M be the $N \times N(N + 1)$ matrix given by

$$M \triangleq [\ M_\infty \ \mid \ M_0 \ \mid \ \cdots \ \mid \ M_z \ \mid \ \cdots\ ]. \tag{30}$$

Note that $M_\infty = I_N$ and $M_0 = H_N$.

Definition 19: A *symmetry* of M is a pair (U, G) such that $UMG = M$, where U is an $N \times N$ unitary matrix, and G is a generalized permutation matrix, i.e., $G = \Pi D$ where Π is a permutation matrix and D is a diagonal matrix of quaternary phases.

Observe that for any such symmetry, G can undo the action of U if and only if U induces a (generalized) permutation on the columns of M. Moreover, since U is unitary it has to preserve inner products, so Corollary 16 implies that U can only permute the bases M_z and permute columns within each basis, or equivalently permute the corresponding maximal commutative subgroups and permute elements within each subgroup, respectively, by conjugation.

Lemma 20: For any symmetry (U, G) of M, the unitary matrix U is an element of the Clifford group Cliff_N.

Proof: A Pauli matrix $E(a, b) \in E([I_m \mid P_z])$ that fixes M_z can be written as $E(a, b) = \sum_{\mathsf{v} \in M_z} \epsilon_{\mathsf{v}} \mathsf{v}\mathsf{v}^\dagger$, where $\epsilon_{\mathsf{v}} = \pm 1$ for all v. Since U permutes the eigenbases M_z, it follows that $U\mathsf{v} \in M_{z'}$, for some $z' \in \mathbb{F}_{2^m} \cup \{\infty\}$, is fixed by $UE(a, b)U^\dagger$ which must again be a Pauli matrix. Hence U is a Clifford element. ■

We first observe the symmetries induced by elements of HW_N.

1) Pauli Matrices $E(a, 0)$ and $E(0, b)$: The group $E([I_m \mid 0])$ of Pauli matrices $E(a, 0)$ fixes each column of $M_0 = H_N$ and acts transitively on the columns of each of the other N blocks. The group $E([0 \mid I_m])$ of Pauli matrices $E(0, b)$ fixes each column of $M_\infty = I_N$ and acts transitively on the columns of each of the remaining blocks.

Definition 21: The *projective special linear group* $\mathrm{PSL}(2, 2^m)$ is the group of all transformations

$$f(z) = \frac{az+b}{cz+d}, \text{ where } a, b, c, d \in \mathbb{F}_{2^m}, ad+bc = 1, \quad (31)$$

acting on the projective line $\mathbb{F}_{2^m} \cup \{\infty\}$. The transformation f is associated with the action of the 2×2 matrix $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ on 1-dimensional spaces, since

$$f\colon \begin{bmatrix} z \\ 1 \end{bmatrix} \mapsto \begin{bmatrix} az+b \\ cz+d \end{bmatrix} \equiv \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ or } \begin{bmatrix} \frac{az+b}{cz+d} \\ 1 \end{bmatrix}.$$

The *projective linear group* $\mathrm{P\Gamma L}(2, 2^m)$ is the group of all transformations

$$f(z) = \frac{az^{2^{-i}}+b}{cz^{2^{-i}}+d}, \text{ where } a, b, c, d \in \mathbb{F}_{2^m}, ad+bc = 1, \quad (32)$$

and $i \in \{0, 1, \ldots, m-1\}$. The orders are

$$|\mathrm{PSL}(2, 2^m)| = (N+1)N(N-1) = 2^{3m} - 2^m,$$
$$|\mathrm{P\Gamma L}(2, 2^m)| = (N+1)N(N-1)m. \quad (33)$$

Now we analyze the symmetries induced by elements of the binary symplectic group $\mathrm{Sp}(2m, \mathbb{F}_2)$.

2) Clifford Symmetries of M: The group $\mathrm{PSL}(2, 2^m)$ is generated by the transformations $z \mapsto z+x, z \mapsto zx$, and $z \mapsto 1/z$. The group $\mathrm{P\Gamma L}(2, 2^m)$ is $\mathrm{PSL}(2, 2^m)$ enlarged by the Frobenius automorphisms $z \mapsto z^{2^{-i}} \equiv zR^{-i}$ discussed in Section IV-A. We realize each of these transformations as a symmetry of M. We recall that $A_z W A_z^T = A_z^2 W$ from part (c) of Lemma 4, and for convenience we work with maximal commutative subgroups $E([I_m \mid A_z^2 W])$, i.e., the Kerdock matrices are $P_z = A_z^2 W$. Note that every field element $\beta \in \mathbb{F}_{2^m}$ is a square, so this is equivalent to $P_z = A_z W$.

(i) $z \mapsto z+x$ becomes $[I_m \mid A_z^2 W] \mapsto [I_m \mid A_{x+z}^2 W]$:

$$[I_m \mid A_z^2 W] \begin{bmatrix} I_m & A_x^2 W \\ 0 & I_m \end{bmatrix} = [I_m \mid (A_z^2 + A_x^2) W]$$
$$\equiv [I_m \mid (A_{x+z}^2) W]. \quad (34)$$

(ii) $z \mapsto xz$ becomes $[I_m \mid A_z^2 W] \mapsto [I_m \mid A_{xz}^2 W]$:

$$[I_m \mid A_z^2 W] \begin{bmatrix} A_x^{-1} & 0 \\ 0 & A_x^T \end{bmatrix} = [A_x^{-1} \mid A_z^2 W A_x^T]$$
$$= [A_x^{-1} \mid A_x A_z^2 W]$$
$$\equiv [I_m \mid A_{xz}^2 W]. \quad (35)$$

(iii) $z \mapsto 1/z$ becomes $[I_m \mid A_z^2 W] \mapsto [I_m \mid A_{z^{-1}}^2 W]$:

$$[I_m \mid A_z^2 W] \begin{bmatrix} 0 & I_m \\ I_m & 0 \end{bmatrix} \begin{bmatrix} W^{-1} & 0 \\ 0 & W^T \end{bmatrix}$$
$$= [A_z^2 W \mid I_m] \begin{bmatrix} W^{-1} & 0 \\ 0 & W \end{bmatrix}$$
$$= [A_z^2 \mid W]$$
$$\equiv [I_m \mid A_{z^{-1}}^2 W]. \quad (36)$$

Note that if we start with $z = 0$, i.e., the subgroup $E([I_m \mid 0])$, then since W is invertible the final subgroup is $E([0 \mid I_m])$, interpreted as $z = \infty$.

(iv) $z \mapsto z' \triangleq z^{2^{-i}}$ becomes $[I_m \mid A_z^2 W] \mapsto [I_m \mid A_{z'}^2 W]$:

$$[I_m \mid A_z^2 W] \begin{bmatrix} R^{-i} & 0 \\ 0 & (R^i)^T \end{bmatrix} = [R^{-i} \mid A_z^2 W (R^i)^T]$$
$$= [R^{-i} \mid A_z^2 R^{-i} W]$$
$$\equiv [I_m \mid A_{z'}^2 W]. \quad (37)$$

Let $\mathfrak{P}_{\mathrm{K},m}$ be the group of symplectic transformations generated by (i), (ii) and (iii) above, and let $\mathfrak{P}_{\mathrm{K},m}^*$ be the group $\mathfrak{P}_{\mathrm{K},m}$ enlarged by the generators (iv). Thus, using notation in Table I, we have

$$\mathfrak{P}_{\mathrm{K},m} \triangleq \langle T_{A_x^2 W},\ L_{A_x^{-1}},\ \Omega L_{W^{-1}}; x \in \mathbb{F}_{2^m} \rangle \cong \mathrm{PSL}(2, 2^m), \quad (38)$$

$$\mathfrak{P}_{\mathrm{K},m}^* \triangleq \langle T_{A_x^2 W},\ L_{R^{-i} A_x^{-1}},\ \Omega L_{W^{-1}}; x \in \mathbb{F}_{2^m} \rangle \cong \mathrm{P\Gamma L}(2, 2^m). \quad (39)$$

Although $T_W^2 = I_{2m}$, in the unitary group we have $t_W^2 = E(0, d_W)$. Therefore the corresponding Clifford elements will generate a group larger than $\mathrm{PSL}(2, 2^m)$ and $\mathrm{P\Gamma L}(2, 2^m)$. Each symplectic matrix in the above groups can be transformed into a quantum circuit (or simply a unitary matrix) by expressing it as a product of standard symplectic matrices from Table I (see [41, Section II]).

Remark 22: Note that $\Omega \notin \mathfrak{P}_{\mathrm{K},m}$ but $\Omega L_{W^{-1}} \in \mathfrak{P}_{\mathrm{K},m}$, which means H_N does not permute columns of M but $H_N \ell_{W^{-1}}$ does. Hence, for example, to map $[0, a]$ to $[a, 0]$ one sequence would be $(\Omega L_{W^{-1}}) \cdot L_{A_b^{-1}}$, where b satisfies $aW^{-1} A_b^{-1} = a$. This does not force $A_b^{-1} = W$.

Lemma 23: Any element of $\mathfrak{P}_{\mathrm{K},m}$ can be described as a product of at most 4 basis symplectic matrices given in Section V-B2.

Proof: Using the results in this Section, a general block permutation $[I_m \mid A_z^2 W] \mapsto [I_m \mid A_{\frac{az+b}{cz+d}}^2 W]$ is realized as

$$[I_m \mid A_z^2 W] \begin{bmatrix} A_d^2 & A_b^2 W \\ W^{-1} A_c^2 & (A_a^2)^T \end{bmatrix} = [A_{cz+d}^2 \mid A_{az+b}^2 W]$$
$$\equiv \left[I_m \mid A_{\frac{az+b}{cz+d}}^2 W\right].$$

It can be verified that the above is a valid symplectic matrix and satisfies all conditions in (10). We now show that this matrix can be decomposed as a product of 4 basis matrices.

We use the results in Lemma 4 and observe the following:

$$\begin{bmatrix} A_x^{-2} & 0 \\ 0 & (A_x^2)^T \end{bmatrix} \begin{bmatrix} 0 & W^T \\ W^{-1} & 0 \end{bmatrix} = \begin{bmatrix} 0 & A_x^{-2} W \\ W^{-1} A_x^2 & 0 \end{bmatrix};$$

$$\begin{bmatrix} I_m & A_y^2 W \\ 0 & I_m \end{bmatrix} \begin{bmatrix} 0 & A_x^{-2} W \\ W^{-1} A_x^2 & 0 \end{bmatrix} = \begin{bmatrix} A_{xy}^2 & A_x^{-2} W \\ W^{-1} A_x^2 & 0 \end{bmatrix};$$

$$\begin{bmatrix} A_{xy}^2 & A_x^{-2} W \\ W^{-1} A_x^2 & 0 \end{bmatrix} \begin{bmatrix} I_m & A_w^2 W \\ 0 & I_m \end{bmatrix} = \begin{bmatrix} A_{xy}^2 & A_{wxy+x^{-1}}^2 W \\ W^{-1} A_x^2 & (A_{wx}^2)^T \end{bmatrix}.$$

Hence we set $x \triangleq c, w \triangleq \frac{a}{c}, y \triangleq \frac{d}{c}$ so that $wxy + x^{-1} = \frac{ad+1}{c} = b$ and the resultant matrix matches the general symplectic matrix given above. ■

Corollary 24: Let $a, b, c, d \in \mathbb{F}_{2^m}$ be such that $ad + bc = 1$. The isomorphism $\tau\colon \mathrm{PSL}(2, 2^m) \to \mathfrak{P}_{\mathrm{K},m}$ can be defined as

$$\tau\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right) \triangleq T_{A_{d/c}^2 W} \cdot L_{A_c^{-2}} \cdot \Omega L_{W^{-1}} \cdot T_{A_{a/c}^2 W}. \quad (40)$$

Observe that this provides a systematic procedure to sample from the group $\mathfrak{P}_{\mathrm{K},m}$. By choosing $\alpha, \beta, \delta \in \mathbb{F}_{2^m}$ uniformly at random, a symmetry element can be constructed as

$$F_{\alpha,\beta,\delta} \triangleq T_{A_\alpha W} \cdot L_{A_\beta} \cdot (\Omega L_{W^{-1}} \cdot T_{A_\delta W}). \tag{41}$$

The first two factors provide transitivity on the Hermitian matrices of all maximal commutative subgroups except $Z_N = E([0 \mid I_m])$, and the last factor enables exchanging any subgroup $E([I_m \mid P_z])$ with $E([0 \mid I_m])$ (see Lemma 31).

We complete this section by observing that the symmetry group can be enlarged by including the Frobenius automorphisms R from Section IV-A.

Lemma 25: An arbitrary element from $\mathfrak{P}^*_{\mathrm{K},m}$ specified by $a, b, c, d \in \mathbb{F}_{2^m}$ and $i \in \{0, \ldots, m-1\}$ takes the form

$$F = \begin{bmatrix} R^{-i} A_d^2 & R^{-i} A_b^2 W \\ W^{-1} R^{-i} A_c^2 & (R^i)^T (A_a^2)^T \end{bmatrix}, \tag{42}$$

with $ad + bc = 1$, and realizes the block permutation

$$[I_m \mid A_z^2 W] \longmapsto \left[I_m \mid A^2_{\frac{az'+b}{cz'+d}} W \right], \; z' \triangleq z^{2^{-i}}. \tag{43}$$

Proof: See Appendix A. ∎

VI. Unitary 2-Designs from the Kerdock MUBs

In this section, we show that the unitary transformations determined by $\mathfrak{P}_{\mathrm{K},m}$, along with Pauli matrices $D(a, b) \in HW_N$, form a unitary 2-design. We first define a graph on Pauli matrices, where Clifford elements act as graph automorphisms. We then show that a group of automorphisms that acts transitively on vertices forms a unitary 2-design. Finally we show that a group of automorphisms that acts transitively on vertices, on edges, and on non-edges forms a unitary 3-design.

Definition 26: The *Heisenberg-Weyl graph* $\mathbb{H}_N$ has $N^2 - 1$ vertices, labeled by pairs $\pm E(a, b)$ with $[a, b] \neq [0, 0]$ where vertices labeled $\pm E(a, b)$ and $\pm E(c, d)$ are joined if $E(a, b)$ commutes with $E(c, d)$. We use $[a, b]$ to represent the vertex labeled $\pm E(a, b)$ and $[a, b]$ — $[c, d]$ to represent an edge between two vertices.

Remark 27: Elements of the Clifford group act by conjugation on HW_N, inducing automorphisms of the graph $\mathbb{H}_N$. We shall distinguish two types of edges in $\mathbb{H}_N$. *Type* 1 edges connect vertices from the same maximal commutative subgroup $E([I_m \mid P_z]), z \in \mathbb{F}_{2^m}$, or from $E([0 \mid I_m])$. *Type* 2 edges connect vertices from different maximal commutative subgroups.

To see that $\text{Aut}(\mathbb{H}_N) = \text{Sp}(2m, \mathbb{F}_2)$, determine a symplectic matrix to reduce an arbitrary graph automorphism to an automorphism π that fixes $[e_i, 0], [0, e_i], i = 1, \ldots, m$, then show that π fixes every vertex. This essentially amounts to solving for a symplectic matrix satisfying a linear system.

Definition 28 ([61, Def. 2.4]): A *strongly regular graph* with parameters (n, t, λ, μ) is a graph with n vertices, where each vertex has degree t, and where the number of vertices joined to a pair of distinct vertices x, y is λ or μ according as x, y are joined or not joined respectively.

Lemma 29: The Heisenberg-Weyl graph $\mathbb{H}_N$ is strongly regular with parameters

$$n = N^2 - 1, \; t = \frac{N^2}{2} - 2, \; \lambda = \frac{N^2}{4} - 3, \; \mu = \frac{N^2}{4} - 1. \tag{44}$$

Proof: A vertex $[c, d]$ joined to a given vertex $[a, b]$ is a solution to $[a, b]\Omega[c, d]^T = 0$ (due to (5)). This is a linear system with a single constraint, and after eliminating the solutions $[0, 0]$ and $[a, b]$ we are left with $t = \frac{N^2}{2} - 2$ distinct vertices $[c, d]$ joined to $[a, b]$.

Given vertices $[a, b], [c, d]$ a vertex $[e, f]$ joined to both $[a, b]$ and $[c, d]$ is a solution to a linear system with two independent constraints. When $[a, b]$ is not joined to $[c, d]$, we only need to eliminate the solution $[0, 0]$. When $[a, b]$ is joined to $[c, d]$ we need to eliminate $[0, 0], [a, b]$ and $[c, d]$. ∎

Remark 30: The number of edges in $\mathbb{H}_N$ is $(N^2 - 1)(\frac{N^2}{2} - 2)$. The number of type-1 edges is $(N + 1)(N - 1)(N - 2)/2$ and the number of type-2 edges is $(N^2 - 1)(\frac{N^2}{2} - N)/2$.

Lemma 31:

(a) The symplectic group $\text{Sp}(2m, \mathbb{F}_2)$ acts transitively on vertices, on edges, and on non-edges of $\mathbb{H}_N$.

(b) The groups $\mathfrak{P}_{\mathrm{K},m}$ and $\mathfrak{P}^*_{\mathrm{K},m}$ act transitively on vertices of $\mathbb{H}_N$.

Proof: Part (a) is well-known in symplectic geometry, and can also be proven by direct calculation using symplectic matrices.

(b) Since $\mathfrak{P}_{\mathrm{K},m}$ acts transitively on maximal commutative subgroups $E([0 \mid I_m]), E([I_m \mid P_z]), z \in \mathbb{F}_{2^m}$ (see (34) and (36)), we need only show that $\mathfrak{P}_{\mathrm{K},m}$ is transitive on a particular subgroup, say $E([I_m \mid 0])$. If $a, b \in \mathbb{F}_{2^m}$ then there exists $c \in \mathbb{F}_{2^m}$ such that $b = ac$, and it follows from (35) that the symplectic matrix $\begin{bmatrix} A_c & 0 \\ 0 & A^T_{c^{-1}} \end{bmatrix}$ maps $[a, 0]$ to $[b, 0]$. ∎

Remark 32: The groups $\mathfrak{P}_{\mathrm{K},m}$ and $\mathfrak{P}^*_{\mathrm{K},m}$ are not transitive on edges of $\mathbb{H}_N$ because they cannot mix type-1 and type-2 edges.

Definition 33 ([11] [62, Chap. 7]): Let k be a positive integer. An ensemble $\mathcal{E} = \{p_i, U_i\}_{i=1}^n$, where the unitary matrix U_i is selected with probability p_i, is said to be a *unitary k-design* if for all linear operators $X \in (\mathbb{C}^N)^{\otimes k}$

$$\sum_{(p,U)\in\mathcal{E}} p \; U^{\otimes k} X (U^\dagger)^{\otimes k} = \int_{\mathbb{U}_N} d\eta(U) \; U^{\otimes k} X (U^\dagger)^{\otimes k}, \tag{45}$$

where $\eta(\cdot)$ denotes the Haar measure on the unitary group $\mathbb{U}_N$. The linear transformations determined by each side of (45) are called *k-fold twirls*. A unitary k-design is defined by the property that the ensemble twirl coincides with the full unitary twirl.

We define the *Kerdock twirl* to be the linear transformation of $(\mathbb{C}^N)^{\otimes 2}$ determined by the uniformly weighted ensemble consisting of $\phi^{-1}(\mathfrak{P}_{\mathrm{K},m})$ along with Pauli matrices $D(a, b)$, where $\phi \colon \text{Cliff}_N / HW_N \to \text{Sp}(2m, \mathbb{F}_2)$ (from Section III). Similarly, we define the 2-fold action (in (45)) of the ensemble determined by $\mathfrak{P}^*_{\mathrm{K},m}$ as the *enlarged Kerdock twirl.*

Definition 34: An ensemble $\mathcal{E} = \{p_i, U_i\}_{i=1}^n$ of Clifford elements U_i is *Pauli mixing* if for every vertex $[a, b]$ the distribution $\{p_i, U_i E(a, b) U_i^\dagger\}$ is uniform over vertices of $\mathbb{H}_N$.

The ensemble $\mathcal{E}$ is *Pauli* 2*-mixing* if it is Pauli mixing and if for every edge (resp. non-edge) $([a,b],[c,d])$ the distribution $\{p_i, (U_iE(a,b)U_i^{\dagger}, U_iE(c,d)U_i^{\dagger})\}$ is uniform over edges (resp. non-edges) of $\mathbb{H}_N$.

Theorem 35: Let G be a subgroup of the Clifford group containing all $D(a,b) \in HW_N$, and let $\mathcal{E} = \{\frac{1}{|G|}, U\}_{U \in G}$ be the ensemble defined by the uniform distribution. If G acts transitively on vertices of $\mathbb{H}_N$ then $\mathcal{E}$ is a unitary 2-design, and if G acts transitively on vertices, edges and non-edges then $\mathcal{E}$ is a unitary 3-design.

Proof: Transitivity means a single orbit so that random sampling from G results in the uniform distribution on vertices, edges, or non-edges. Hence transitivity on vertices implies $\mathcal{E}$ is Pauli mixing and transitivity on vertices, edges and non-edges implies $\mathcal{E}$ is Pauli 2-mixing. It now follows from [11] or [16] that Pauli mixing (resp. Pauli 2-mixing) implies $\mathcal{E}$ is a unitary 2-design (resp. unitary 3-design). ■

Corollary 36: Random sampling from the Clifford group gives a unitary 3-design and random sampling from the groups $\mathfrak{P}_{\mathrm{K},m}, \mathfrak{P}^*_{\mathrm{K},m}$ followed by a random Pauli matrix $D(a,b)$ gives unitary 2-designs.

Puchała and Miszczak have developed a useful Mathematica® package `IntU` [63] for symbolic integration with respect to the Haar measure on unitaries. For small m, we used this utility to verify the equality in (45) explicitly for the Kerdock 2-design.

Sampling uniformly from the groups $\mathfrak{P}_{\mathrm{K},m}$ can be achieved using the *systematic* procedure shown in (41). The resultant symplectic matrix can be transformed into a quantum circuit (or simply a unitary matrix) by expressing it as a product of standard symplectic matrices from Table I (see [41, Section II]). Although our unitary 2-design is equivalent to that discovered by Cleve et al. [16], the methods we use to translate design elements to circuits are very different and much simpler. While they use sophisticated methods from finite fields to propose a circuit translation that is *tailored* for the design, our algorithm from [35] (whose details are discussed in [41, Appendix II]) is a *general purpose* procedure that can be used to translate *arbitrary* symplectic matrices to circuits. They have been able to show Clifford-gate-complexity $O(m \log m \log \log m)$ assuming the extended Riemann hypothesis is true, or $O(m \log^2 m \log \log m)$ unconditionally, both of which are *near-linear* when compared to the $O(m^2)$ gate-complexity for general Clifford elements. Our open-source implementation[1] allows one to construct the design for a specified number of qubits, and we use this utility to calculate *worst-case* gate complexities on up to 16 qubits.

In our sampling procedure (41) we have three elementary forms $T_{A_\alpha W}, L_{A_\beta}$, and Ω, which translate to phase and controlled-Z gates, permutations and controlled-NOT gates, and Hadamard gates on all qubits, respectively (see [41, Appendix I]). Note that $L_{W^{-1}}$ has the same elementary form as L_{A_β}, although W is fixed for a given m. The Hadamard gates add only $O(m)$ complexity. Figures 2, 3, and 4 plot the *worst-case* complexities of the gates $T_{A_\alpha W}, L_{A_\beta}$, and $L_{W^{-1}}$ obtained using our procedure. The only form that seems to grow faster than $O(m \log m \log \log m)$ is L_{A_β}, and we are currently investigating methods to calculate this gate complexity via analytical arguments that leverage results in the classical computation literature. A curious data point is $m = 15$ in Fig. 4, where the matrix W has zeros everywhere except the anti-diagonal, which translates to a single permutation of the qubits. Since the decomposition in (41) involves a *constant* number of factors, the overall complexity is that of the factor with largest order term. We will also investigate if our circuits can always be organized to give a depth of $O(\log m)$ just as Cleve et al.

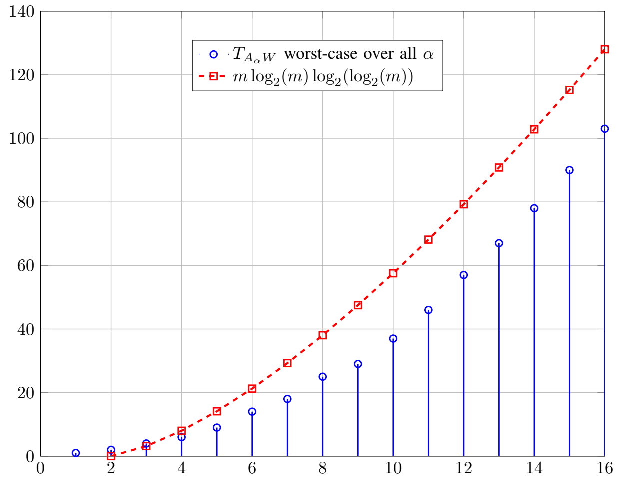


Fig. 2. The gate complexities for the element $T_{A_\alpha W}$ in (41) for varying m.

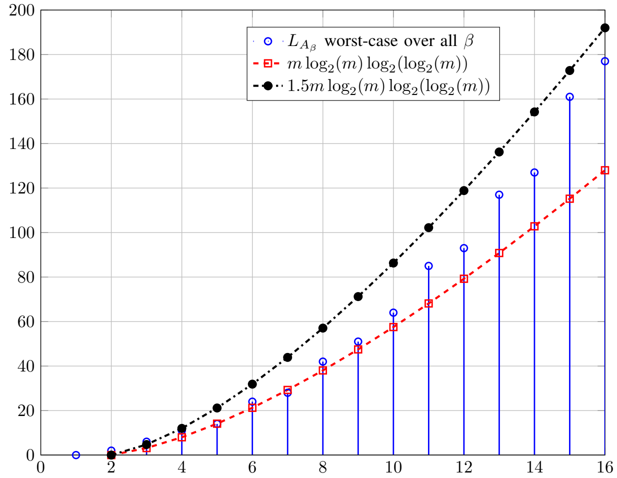


Fig. 3. The gate complexities for the element L_{A_β} in (41) for varying m.

Hence, we have provided an alternative perspective to the *quantum* unitary 2-design discovered by Cleve et al. by establishing a connection to *classical* Kerdock codes, and simplified the description of the design as well as its translation to circuits. Since we also appear to achieve competetive Clifford-gate-complexities, and provide implementations for our methods, we believe this paves the way for employing this 2-design in several applications, specifically in randomized benchmarking [9], [51].

[1] Implementations online: https://github.com/nrenga/symplectic-arxiv18a

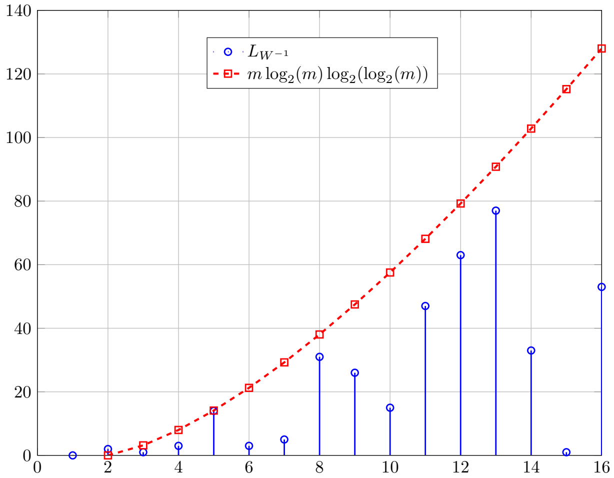


Fig. 4. The gate complexities for the element $L_{W^{-1}}$ in (41) for varying m.

VII. Logical Unitary 2-Designs

In this section, we apply our synthesis algorithm [41] to logical randomized benchmarking, which was recently proposed by Combes et al. [10] as a more precise protocol to reliably estimate performance metrics for an error correction implementation, as compared to the standard approach of physical randomized benchmarking. Using this procedure, they are able to quantify the effects of imperfect logical gates, crosstalk, and correlated errors, which are typically ignored. They use the full logical Clifford group to perform benchmarking as this group forms a (logical) unitary 2-design (as well as 3-design). Our construction can be used to implement their protocol with a much smaller 2-design ($\mathfrak{P}_{\mathrm{K},m}$) and our synthesis algorithm can be used to realize the design at the logical level.

Here we show by example that we can efficiently translate *any* unitary 2-design of Clifford elements into a *logical* unitary 2-design, i.e., for a given $[\![m, m-k]\!]$ stabilizer code, produce m-qubit physical Clifford circuits that form a unitary 2-design on the $m-k$ protected qubits. As a proof of concept, we translate the Kerdock design $\mathfrak{P}_{\mathrm{K},m-k}$ on the $m-k = 4$ logical qubits of the $[\![6,4,2]\!]$ CSS code [52], [64] into their physical 6-qubit implementations. The stabilizers of this code are

$$\begin{aligned} S &\triangleq \langle X^{\otimes 6}, Z^{\otimes 6} \rangle \\ &= \langle E(111111, 000000), E(000000, 111111) \rangle. \end{aligned} \tag{46}$$

The logical Pauli operators are

$$\begin{array}{l|l} \bar{X}_1 \triangleq E(110000, 000000) & \bar{Z}_1 \triangleq E(000000, 010001) \\ \bar{X}_2 \triangleq E(101000, 000000) & \bar{Z}_2 \triangleq E(000000, 001001) \\ \bar{X}_3 \triangleq E(100100, 000000) & \bar{Z}_3 \triangleq E(000000, 000101) \\ \bar{X}_4 \triangleq E(100010, 000000) & \bar{Z}_4 \triangleq E(000000, 000011) \end{array} . \tag{47}$$

Consider $\mathbb{F}_{16}$ constructed by adjoining to $\mathbb{F}_2$ a root α of the primitive polynomial $p(x) = x^4 + x + 1$. Consider the element F_{abcd} in $\mathfrak{P}_{\mathrm{K},4}$ identified by the tuple $(a = \alpha^3, b = \alpha^8, c = \alpha^7, d = 0)$, or equivalently $(a, b, c, d, i = 0)$ in $\mathfrak{P}^*_{\mathrm{K},4}$. In this case the matrices defined in Section IV-A are

$$\begin{aligned} W &= \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}, W^{-1} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \\ R &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}. \end{aligned} \tag{48}$$

Then, using the isomorphism in Corollary 24 and the direct form in Lemma 23, we can express the element in $\mathfrak{P}_{\mathrm{K},4}$ as

$$\begin{aligned} F_{abcd} &= T_{A_0^2 W} \cdot L_{A_{\alpha^7}^{-2}} \cdot \Omega L_{W^{-1}} \cdot T_{A_{\alpha^{11}}^2 W} \\ &= I_8 \cdot L_{(A^7)^{-2}} \cdot \Omega L_{W^{-1}} \cdot T_{(A^{11})^2 W} \qquad (49) \\ &= \begin{bmatrix} 0 & A_{\alpha^8}^2 W \\ W^{-1} A_{\alpha^7}^2 & (A_{\alpha^3}^2)^T \end{bmatrix} \\ &= \begin{bmatrix} 0 & A^{16} W \\ W^{-1} A^{14} & (A^6)^T \end{bmatrix} \qquad (50) \\ &= \left[\begin{array}{cccc|cccc} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ \hline 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{array}\right]. \qquad (51) \end{aligned}$$

Using the explicit decomposition in (49), we map the elementary symplectic matrices to standard Clifford gates (as discussed in [41]) and get the following circuit (`CKT1`).[2]

'Permute'	[4,1,2,3]
'CNOT'	[1,2]
'Permute'	[4,3,2,1]
'CNOT'	[1,4]
'H'	[1,2,3,4]
'P'	[1,2,3,4]
'CZ'	[1,3]
'CZ'	[2,4]
'CZ'	[3,4]

Here the indices corresponding to 'Permute' imply the cycle permutation (1432), i.e., the first qubit has been replaced by the fourth, the fourth by the third, the third by the second, and the second by the first. (Note that we do not simplify circuits to their optimal form here but simply report the results of our synthesis algorithm.) An alternative procedure is to directly input the final symplectic matrix (51) to the symplectic decomposition algorithm in [35] (also see [41, Section II]), yielding the following circuit (`CKT2`).

[2] We list circuits instead of giving their circuit representation to align with the `MATLAB`® cell array format we adopt in our implementations.

‘Permute’	[3,2,1,4]
‘CNOT’	[4,3]
‘CNOT’	[1,4]
‘H’	[1,2,3,4]
‘P’	[1,2,3,4]
‘CZ’	[1,3]
‘CZ’	[2,4]
‘CZ’	[3,4]

The difference in depth of the two circuits is very small in this case, but we found that for about half of the elements in $\mathfrak{P}_{\mathrm{K},4}$ the explicit form in Corollary 24 had smaller depth, while for those remaining, the direct decomposition was better.

Next we translate this logical circuit into its physical implementation for the $[\![6,4,2]\!]$ CSS code. We apply our synthesis algorithm [41], which can be summarized as follows.

1) Compute the action of either of the above circuits on the Pauli matrices X_i, Z_i, for $i = 1, 2, 3, 4$, under conjugation, i.e., compute $gE(e_i, 0)g^{\dagger}, gE(0, e_i)g^{\dagger}$ where g represents the circuit.
2) Translate these into logical constraints on the desired physical circuit $\bar{g} \in \text{Cliff}_{2^6}$ by interpreting X_i, Z_i above as their logical equivalents $\bar{X}_i, \bar{Z}_i \in HW_{2^6}$.
3) Rewrite the above conditions as linear constraints on the desired symplectic matrix $F_{\bar{g}}$ using (9). Add constraints to normalize (or just centralize) the stabilizer S.
4) Solve for all symplectic solutions, compute their corresponding circuits and identify the best solution in terms of smallest depth, with respect to the decomposition in [35] (also discussed in [41, Section II]).
5) Verify the constraints imposed in step 2 and check for any sign violations (due to signs in (9)). In case of violations, identify a Pauli matrix to fix the signs.

Using this algorithm, we computed the circuit with smallest depth (relative to other solutions decomposed using the same algorithm) and obtained the following solution (`CKT3`).

‘Permute’	[1,6,3,5,4,2]
‘CNOT’	[3,1]
‘CNOT’	[4,1]
‘CNOT’	[5,1]
‘CNOT’	[6,1]
‘CNOT’	[6,4]
‘CNOT’	[6,5]
‘CNOT’	[4,5]
‘CNOT’	[3,6]
‘CNOT’	[2,3]
‘H’	[1,2,3,4,5,6]
‘P’	[3,4,5]
‘CZ’	[1,3]
‘CZ’	[1,4]
‘CZ’	[1,5]
‘CZ’	[2,6]
‘CZ’	[3,5]
‘CZ’	[4,5]
‘CZ’	[4,6]
‘CZ’	[5,6]
‘H’	[1,2]
‘CNOT’	[2,3]
‘CNOT’	[2,4]
‘CNOT’	[2,5]
‘CNOT’	[2,6]
‘Z’	[2,6]

We used the same procedure to translate all 4080 elements of $\mathfrak{P}_{\mathrm{K},4}$ into their (smallest depth) physical implementations for the $[\![6,4,2]\!]$ code, and the synthesis took about 25 minutes on a laptop running the Windows 10 operating system (64-bit) with an Intel® Core™ i7-5500U @ 2.40GHz processor and 8GB RAM. Note that for this case each element of $\mathfrak{P}_{\mathrm{K},4}$ translates to $2^{k(k+1)/2} = 8$ symplectic solutions during the above procedure, and we need to compare depth after calculating circuits for all solutions. In future work, we will try to optimize directly for depth without computing all solutions as this procedure is expensive for codes with large redundancy (k). However, since this translation needs to be done only once for a given set $\mathfrak{P}_{\mathrm{K},m-k}$ and an $[\![m, m-k]\!]$ stabilizer code, the circuits can be precomputed and stored in memory.

VIII. Conclusion

In this paper, we provided a simpler calculation of the Hamming weight distribution of the classical binary non-linear Kerdock codes by appealing to the connection with maximal commutative subgroups of the Heisenberg-Weyl group from the quantum domain. Using this connection, we also described the group of Clifford symmetries of the mutually unbiased bases determined by the Kerdock sets of symmetric matrices. We then used this result in the classical domain to construct small unitary 2-designs. Finally, we demonstrated an efficient procedure for translating any unitary 2-design consisting of Clifford elements into a logical unitary 2-design for a given stabilizer code. This work demonstrates yet again that interactions between the classical and quantum domains are still mutually beneficial, both theoretically and practically.

References

[1] S. Bravyi, D. Gosset, and R. König, “Quantum advantage with shallow circuits.,” *Science*, vol. 362, no. 6412, pp. 308–311, 2018.

[2] E. Conover, “Quantum computers are about to get real,” *Science News*, vol. 191, no. 13, p. 28, 2017. https://www.sciencenews.org/article/quantum-computers-are-about-get-real.

[3] P. W. Shor, “Scheme for reducing decoherence in quantum computer memory,” *Phys. Rev. A*, vol. 52, no. 4, pp. R2493–R2496, 1995.

[4] A. R. Calderbank and P. W. Shor, “Good quantum error-correcting codes exist,” *Phys. Rev. A*, vol. 54, pp. 1098–1105, Aug 1996.

[5] A. M. Steane, “Simple quantum error-correcting codes,” *Phys. Rev. A*, vol. 54, no. 6, pp. 4741–4751, 1996.

[6] R. Calderbank, E. Rains, P. Shor, and N. Sloane, “Quantum error correction via codes over GF(4),” *IEEE Trans. Inform. Theory*, vol. 44, pp. 1369–1387, Jul 1998.

[7] D. Gottesman, *Stabilizer codes and quantum error correction.* PhD thesis, California Institute of Technology, 1997.

[8] J. Emerson, R. Alicki, and K. Życzkowski, “Scalable noise estimation with random unitary operators,” *J. Opt. B Quantum Semiclassical Opt.*, vol. 7, no. 10, pp. S347–S352, 2005.

[9] E. Magesan, J. M. Gambetta, and J. Emerson, “Characterizing quantum gates via randomized benchmarking,” *Phys. Rev. A*, vol. 85, no. 4, p. 042311, 2012.

[10] J. Combes, C. Granade, C. Ferrie, and S. T. Flammia, "Logical Randomized Benchmarking," *arXiv preprint arXiv:1702.03688*, 2017. [Online]. Available: http://arxiv.org/abs/1702.03688.

[11] Z. Webb, "The Clifford group forms a unitary 3-design," *Quantum Inf. Comput.*, vol. 16, no. 15-16, pp. 1379—-1400, 2016. [Online]. Available: https://arxiv.org/pdf/1510.02769.pdf.

[12] D. Gross, K. Audenaert, and J. Eisert, "Evenly distributed unitaries: On the structure of unitary designs," *J. Math. Phys.*, vol. 48, no. 5, p. 052104, 2007.

[13] A. Roy and A. J. Scott, "Unitary designs and codes," *Designs, Codes Cryptogr.*, vol. 53, no. 1, pp. 13–31, 2009.

[14] A. W. Harrow and R. A. Low, "Random quantum circuits are approximate 2-designs," *Commun. in Math. Phy.*, vol. 291, no. 1, pp. 257–302, 2009.

[15] Y. Nakata, C. Hirche, C. Morgan, and A. Winter, "Unitary 2-designs from random X- and Z-diagonal unitaries," *J. Math. Phys.*, vol. 58, no. 5, p. 052203, 2017.

[16] R. Cleve, D. Leung, L. Liu, and C. Wang, "Near-linear constructions of exact unitary 2-designs," *Quantum Inf. Comput.*, vol. 16, no. 9-10, pp. 0721–0756, 2016. [Online]. Available: https://arxiv.org/pdf/1501.04592.pdf.

[17] D. DiVincenzo, D. Leung, and B. Terhal, "Quantum Data Hiding," *IEEE Trans. Inform. Theory*, vol. 48, no. 3, pp. 580–598, 2002.

[18] C. Dankert, R. Cleve, J. Emerson, and E. Livine, "Exact and approximate unitary 2-designs and their application to fidelity estimation," *Phys. Rev. A*, vol. 80, no. 1, p. 012304, 2009.

[19] A. J. Scott, "Optimizing quantum process tomography with unitary 2-designs," *arXiv preprint arXiv:0711.1017*, 2007. [Online]. Available: http://arxiv.org/abs/0711.1017.

[20] M. Quadeer, M. Tomamichel, and C. Ferrie, "Minimax quantum state estimation under Bregman divergence," *Quantum*, vol. 3, p. 126, 2019. [Online]. Available: http://arxiv.org/abs/1808.08984.

[21] P. Hayden, M. Horodecki, A. Winter, and J. Yard, "A decoupling approach to the quantum capacity," *arXiv preprint arXiv:quant-ph/0702005*, 2007. [Online]. Available: http://arxiv.org/abs/quant-ph/0702005.

[22] O. Szehr, F. Dupuis, M. Tomamichel, and R. Renner, "Decoupling with unitary approximate two-designs," *New J. Phys.*, vol. 15, no. 5, p. 053022, 2013.

[23] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. North-Holland, Amsterdam, 1977.

[24] A. R. Hammons, P. V. Kumar, A. R. Calderbank, N. J. A. Sloane, and P. Solé, "The $\mathbb{Z}_4$-Linearity of Kerdock, Preparata, Goethals, and Related Codes," *IEEE Trans. Inform. Theory*, vol. 40, no. 2, pp. 301–319, 1994.

[25] H. Weyl, *The Classical Groups: Their Invariants and Representations*, vol. 1. Princeton University Press, 1997.

[26] R. Calderbank and S. Jafarpour, "Reed Muller Sensing Matrices and the LASSO," *arXiv preprint arXiv:1004.4949*, Apr 2010. [Online]. Available: http://arxiv.org/abs/1004.4949.

[27] J. Dehaene and B. De Moor, "Clifford group, stabilizer states, and linear and quadratic operations over GF(2)," *Phys. Rev. A*, vol. 68, p. 042318, Oct 2003.

[28] S. Aaronson and D. Gottesman, "Improved simulation of stabilizer circuits," *Phys. Rev. A*, vol. 70, no. 5, p. 052328, 2004.

[29] K. Koike and I. Terada, "Young-diagrammatic methods for the representation theory of the classical groups of type Bn, Cn, Dn," *J. Algebr.*, vol. 107, pp. 466–511, May 1987.

[30] R. Kueng and D. Gross, "Qubit stabilizer states are complex projective 3-designs," *arXiv preprint arXiv:1510.02767*, 2015. [Online]. Available: https://arxiv.org/abs/1510.02767.

[31] R. Kueng, H. Zhu, and D. Gross, "Low rank matrix recovery from Clifford orbits," *arXiv preprint arXiv:1610.08070*, 2016. [Online]. Available: https://arxiv.org/abs/1610.08070.

[32] A. Thompson and R. Calderbank, "Compressed Neighbour Discovery using Sparse Kerdock Matrices," in *Proc. IEEE Int. Symp. Inform. Theory*, pp. 2286–2290, Jun 2018. [Online]. Available: http://arxiv.org/abs/1801.04537.

[33] A. Calderbank, P. Cameron, W. Kantor, and J. Seidel, "$\mathbb{Z}_4$-Kerdock Codes, Orthogonal Spreads, and Extremal Euclidean Line-Sets," *Proc. London Math. Soc.*, vol. 75, pp. 436–480, Sep 1997.

[34] O. Tirkkonen, C. Boyd, and R. Vehkalahti, "Grassmannian codes from multiple families of mutually unbiased bases," in *Proc. IEEE Int. Symp. Inform. Theory*, pp. 789–793, Jun 2017.

[35] T. Can, "The Heisenberg-Weyl group, finite symplectic geometry, and their applications," Senior Thesis, Duke University, May 2018.

[36] D. Maslov and M. Roetteler, "Shorter Stabilizer Circuits via Bruhat Decomposition and Quantum Circuit Transformations," *IEEE Trans. Inform. Theory*, vol. 64, no. 7, pp. 4729–4738, 2018.

[37] P. Diaconis and M. Shahshahani, "The subgroup algorithm for generating uniform random variables," *Prob. Eng. Inf. Sci.*, vol. 1, no. 1, pp. 15–32, 1987.

[38] R. Koenig and J. A. Smolin, "How to efficiently select an arbitrary Clifford group element," *J. Math. Phys.*, vol. 55, p. 122202, Dec 2014.

[39] K. Choromanski, F. Fagan, C. Gouy-Pailler, A. Morvan, T. Sarlos, and J. Atif, "TripleSpin - a generic compact paradigm for fast machine learning computations," *arXiv preprint arXiv:1605.09046*, 2016. [Online]. Available: http://arxiv.org/abs/1605.09046.

[40] P. W. Jones, A. Osipov, and V. Rokhlin, "A randomized approximate nearest neighbors algorithm," *Appl. Comput. Harmon. Anal.*, vol. 34, no. 3, pp. 415–444, 2013.

[41] N. Rengaswamy, R. Calderbank, S. Kadhe, and H. D. Pfister, "Synthesis of Logical Clifford Operators via Symplectic Geometry," in *Proc. IEEE Int. Symp. Inform. Theory*, pp. 791–795, Jun 2018. [Online]. Available: http://arxiv.org/abs/1803.06987.

[42] J. T. Anderson, G. Duclos-Cianci, and D. Poulin, "Fault-Tolerant Conversion between the Steane and Reed-Muller Quantum Codes," *Phys. Rev. Lett.*, vol. 113, no. 8, p. 080501, 2014. [Online]. Available: http://arxiv.org/abs/1403.2734.

[43] A. W. Nordstrom and J. P. Robinson, "An optimum nonlinear code," *Inform. and Control*, vol. 11, no. 5-6, pp. 613–616, 1967.

[44] A. Kerdock, "A class of low-rate nonlinear binary codes," *Inform. and Control*, vol. 20, pp. 182–187, Mar 1972.

[45] F. P. Preparata, "A class of optimum nonlinear double-error correcting codes," *Inform. and Control*, vol. 13, no. 13, pp. 378–400, 1968.

[46] J. Goethals, "Two dual families of nonlinear binary codes," *Electronic Letters*, vol. 10, no. 23, p. 471, 1974.

[47] J.-M. Goethals, "Nonlinear codes defined by quadratic forms over GF(2)," *Inform. and Control*, vol. 31, pp. 43–74, May 1976.

[48] P. Delsarte and J. M. Goethals, "Alternating bilinear forms over GF(q)," *J. Comb. Theory, Ser. A*, vol. 19, no. 1, pp. 26–50, 1975.

[49] W. C. Huffman and V. Pless, *Fundamentals of Error-Correcting Codes*. Cambridge University Press, 2003.

[50] A. Thompson and R. Calderbank, "Sparse near-equiangular tight frames with applications in full duplex wireless communication," in *IEEE Glob. Conf. Signal Inf. Process.*, pp. 868–872, Nov 2017.

[51] J. Helsen, J. J. Wallman, S. T. Flammia, and S. Wehner, "Multi-qubit Randomized Benchmarking Using Few Samples," *arXiv preprint arXiv:1701.04299*, 2017. [Online]. Available: http://arxiv.org/abs/1701.04299.

[52] D. Gottesman, "A Theory of Fault-Tolerant Quantum Computation," *arXiv preprint arXiv:quant-ph/9702029*, 1997. [Online]. Available: http://arxiv.org/pdf/quant-ph/9702029.pdf.

[53] D. Gottesman, "An Introduction to Quantum Error Correction and Fault-Tolerant Quantum Computation," *arXiv preprint arXiv:0904.2557*, 2009. [Online]. Available: http://arxiv.org/abs/0904.2557.

[54] M. M. Wilde, *Quantum Information Theory*. Cambridge University Press, 2013.

[55] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.

[56] M. A. De Gosson, *Symplectic Geometry and Quantum Mechanics*, vol. 166. Springer Science & Business Media, 2006.

[57] C. Chamberland, P. Iyer, and D. Poulin, "Fault-tolerant quantum computing in the Pauli or Clifford frame with slow error diagnostics," *Quantum*, vol. 2, p. 43, 2018. [Online]. Available: https://arxiv.org/pdf/1704.06662.pdf.

[58] R. Calderbank, S. Howard, S. Jafarpour, and J. Kent, "Sparse approximation and compressed sensing using the Reed-Muller sieve," Tech. Rep. TR-888-10, Department of Computer Science, Princeton University, 2010.

[59] R. J. McEliece, *Finite Fields for Computer Scientists and Engineers*. Kluwer Academic Publishers, 1987.

[60] A. Calderbank, R. Hardin, E. Rains, P. Shor, and N. Sloane, "A Group-Theoretic Framework for the Construction of Packings in Grassmannian Spaces," *J. Algebr. Comb.*, vol. 9, no. 2, pp. 129–140, 1999.

[61] P. J. Cameron and J. H. Van Lint, *Designs, Graphs, Codes and their Links*, vol. 3. Cambridge University Press, 1991.

[62] J. Watrous, *The Theory of Quantum Information*. Cambridge University Press, 2018.

[63] Z. Puchała and J. A. Miszczak, "Symbolic integration with respect to the haar measure on the unitary groups," *Bulletin of the Polish Academy of Sciences Technical Sciences*, vol. 65, no. 1, pp. 21–27, 2017.

[64] R. Chao and B. W. Reichardt, "Fault-tolerant quantum computation with few qubits," *arXiv preprint arXiv:1705.05365*, 2017. [Online]. Available: http://arxiv.org/pdf/1705.05365.pdf.

Appendix A
Proof of Lemma 25

We proceed as in the proof of Lemma 23 to derive the general form of an element in $\mathfrak{P}^{*}_{\mathrm{K},m}$. Introducing the new generators $L_{R^{-i}}$, and using identities from Lemma 4, we calculate

$$\begin{aligned}
&\begin{bmatrix} R^{-i}A_x^{-2} & 0 \\ 0 & (R^i)^T(A_x^2)^T \end{bmatrix} \begin{bmatrix} 0 & W^T \\ W^{-1} & 0 \end{bmatrix} \\
&= \begin{bmatrix} 0 & R^{-i}A_x^{-2}W \\ (R^i)^T W^{-1} A_x^2 & 0 \end{bmatrix}; \\
&\begin{bmatrix} I_m & A_y^2 W \\ 0 & I_m \end{bmatrix} \begin{bmatrix} 0 & R^{-i}A_x^{-2}W \\ (R^i)^T W^{-1} A_x^2 & 0 \end{bmatrix} \\
&= \begin{bmatrix} A_y^2 W (R^i)^T W^{-1} A_x^2 & R^{-i}A_x^{-2}W \\ (R^i)^T W^{-1} A_x^2 & 0 \end{bmatrix}; \\
&\begin{bmatrix} A_y^2 W (R^i)^T W^{-1} A_x^2 & R^{-i}A_x^{-2}W \\ (R^i)^T W^{-1} A_x^2 & 0 \end{bmatrix} \begin{bmatrix} I_m & A_w^2 W \\ 0 & I_m \end{bmatrix} \\
&= \begin{bmatrix} A_y^2 (W (R^i)^T W^{-1}) A_x^2 & A_y^2 (W (R^i)^T W^{-1}) A_x^2 A_w^2 W + R^{-i}A_x^{-2}W \\ (R^i)^T W^{-1} A_x^2 & (R^i)^T W^{-1} A_x^2 A_w^2 W \end{bmatrix} \\
&= \begin{bmatrix} A_y^2 R^{-i} A_x^2 & A_y^2 R^{-i} A_{wx}^2 W + R^{-i}A_x^{-2}W \\ W^{-1} R^{-i} A_x^2 & (R^i)^T (A_{wx}^2)^T \end{bmatrix} \\
&\triangleq F.
\end{aligned}$$

In this case the relations between a, b, c, d and x, y, z, w that will yield the desired map are unclear. Hence, we first determine the transformation on $[I_m \mid A_z^2 W]$ in terms of x, y, z and w. Again, we repeatedly invoke identities from Lemma 4.

$$\begin{aligned}
&[I_m \mid A_z^2 W]F \\
&= [A_y^2 R^{-i} A_x^2 + A_z^2 R^{-i} A_x^2 \mid (A_y^2 R^{-i} A_{wx}^2 + R^{-i} A_x^{-2})W \\
&\qquad\qquad + A_z^2 W (R^i)^T (A_{wx}^2)^T] \\
&= [A_{y+z}^2 R^{-i} A_x^2 \mid (A_{y+z}^2 R^{-i} A_{wx}^2 + R^{-i} A_x^{-2})W],
\end{aligned}$$

where we have simplified the last term as

$$A_z^2 W (R^i)^T (A_{wx}^2)^T = A_z^2 R^{-i} W (A_{wx}^2)^T = A_z^2 R^{-i} A_{wx}^2 W.$$

Now we have the following simplifications for the three terms.

$$\begin{aligned}
A_{y+z}^2 R^{-i} A_x^2 &= A_{y+z}^2 A_x^{2^{i+1}} R^{-i}, \\
A_{y+z}^2 R^{-i} A_{wx}^2 &= A_{y+z}^2 A_{wx}^{2^{i+1}} R^{-i}, \\
R^{-i} A_x^{-2} &= A_x^{-2^{i+1}} R^{-i}.
\end{aligned}$$

Applying this back we get

$$\begin{aligned}
&[I_m \mid A_z^2 W]F \\
&= \left[A_{y+z}^2 A_x^{2^{i+1}} R^{-i} \mid \left(A_{y+z}^2 A_{wx}^{2^{i+1}} + A_x^{-2^{i+1}}\right) R^{-i} W\right] \\
&\equiv \left[I_m \mid R^i A_x^{-2^{i+1}} A_{y+z}^{-2} \left(A_{y+z}^2 A_{wx}^{2^{i+1}} + A_x^{-2^{i+1}}\right) R^{-i} W\right] \\
&= \left[I_m \mid R^i \left(A_w^{2^{i+1}} + A_{y+z}^{-2} A_x^{-2 \cdot 2^{i+1}}\right) R^{-i} W\right] \\
&= \left[I_m \mid A_w^2 W + R^i A_{y+z}^{-2} A_x^{-2 \cdot 2^{i+1}} R^{-i} W\right] \\
&= \left[I_m \mid A_w^2 W + A_{y+z}^{-2^{1-i}} R^i A_x^{-2 \cdot 2^{i+1}} R^{-i} W\right] \\
&= \left[I_m \mid A_w^2 W + A_{y+z}^{-2^{1-i}} \left(R^i A_{x^{2^{i+1}}}^{-2} R^{-i}\right) W\right] \\
&= \left[I_m \mid A_w^2 W + A_{y+z}^{-2^{1-i}} A_{x^{2^{i+1}}}^{-2^{1-i}} W\right] \\
&= \left[I_m \mid \left(A_w^2 + A_{(y+z)^{-2^{-i}} x^{-2}}^2\right) W\right] \\
&= \left[I_m \mid A_{w+(y+z)^{-2^{-i}} x^{-2}}^2 W\right].
\end{aligned}$$

Now we define $x \triangleq c, w \triangleq \frac{a}{c}$ and $y \triangleq \left(\frac{d}{c}\right)^{2^i}$. Then we get

$$\begin{aligned}
w + (y+z)^{-2^{-i}} x^{-2} &= \frac{a}{c} + \left(z + \left(\frac{d}{c}\right)^{2^i}\right)^{-2^{-i}} \frac{1}{c^2} \\
&= \frac{a}{c} + \left(z^{2^{-i}} + \left(\frac{d}{c}\right)^{2^i \cdot 2^{-i}}\right)^{-1} \frac{1}{c^2} \\
&= \frac{1}{c}\left[a + \frac{1}{cz^{2^{-i}} + c \cdot \frac{d}{c}}\right] \\
&= \frac{1}{c}\left[\frac{acz^{2^{-i}} + ad + 1}{cz^{2^{-i}} + d}\right] \\
&= \frac{az^{2^{-i}} + \left(\frac{ad+1}{c}\right)}{cz^{2^{-i}} + d} \\
&= \frac{az^{2^{-i}} + b}{cz^{2^{-i}} + d}\ ;\ b = \frac{ad+1}{c}.
\end{aligned}$$

Hence we have proved that F performs the permutation

$$[I_m \mid A_z^2 W] \longmapsto \left[I_m \mid A_{\frac{az'+b}{cz'+d}}^2 W\right],\ z' \triangleq z^{2^{-i}}.$$

We note that the above definitions for x, w, y also satisfy the special case of $i = 0$ that corresponds to the proof in Lemma 23. We now substitute these back in F and observe the following simplifications.

$$\begin{aligned}
(A_y^2 R^{-i}) A_x^2 = R^{-i} A_y^{2^{-i+1}} A_x^2 &= R^{-i} A_{y^{2^{-i}} x}^2 = R^{-i} A_d^2, \\
(A_y^2 R^{-i}) A_{wx}^2 + R^{-i} A_x^{-2} &= R^{-i} A_{d/c}^2 A_a^2 + R^{-i} A_{c^{-1}}^2 = R^{-i} A_b^2, \\
W^{-1} R^{-i} A_x^2 &= W^{-1} R^{-i} A_c^2 = (R^i)^T W^{-1} A_c^2, \\
(R^i)^T (A_{wx}^2)^T &= (R^i)^T (A_a^2)^T.
\end{aligned}$$

These imply that the general form of an element in $\mathfrak{P}^{*}_{\mathrm{K},m}$ is

$$\begin{aligned}
F &= \begin{bmatrix} A_y^2 R^{-i} A_x^2 & (A_y^2 R^{-i} A_{wx}^2 + R^{-i} A_x^{-2})W \\ W^{-1} R^{-i} A_x^2 & (R^i)^T (A_{wx}^2)^T \end{bmatrix} \\
&= \begin{bmatrix} R^{-i} A_d^2 & R^{-i} A_b^2 W \\ W^{-1} R^{-i} A_c^2 & (R^i)^T (A_a^2)^T \end{bmatrix}. \qquad \blacksquare
\end{aligned}$$