> "A little misunderstanding? Galileo and the Pope had a little misunderstanding..."

— Sheldon Cooper, The Big Bang Theory.

Generator Matrix Based Search for Extremal Self-Dual Binary Error-Correcting Codes

Martin Derka

Computer Science

Submitted in partial fulfillment
of the requirements for the degree of

Master of Science

Faculty of Computer Science, Brock University
St. Catharines, Ontario

To my family and Stephanie.

Abstract

Self-dual doubly even linear binary error-correcting codes, often referred to as Type II codes, are codes closely related to many combinatorial structures such as 5-designs. Extremal codes are codes that have the largest possible minimum distance for a given length and dimension.

The existence of an extremal $(72, 36, 16)$ Type II code is still open. Previous results show that the automorphism group of a putative code $\mathcal{C}$ with the aforementioned properties has order 5 or dividing 24. In this work, we present a method and the results of an exhaustive search showing that such a code $\mathcal{C}$ cannot admit an automorphism group $\mathbb{Z}_6$.

In addition, we present so far unpublished construction of the extended Golay code by P. Becker. We generalize the notion and provide example of another Type II code that can be obtained in this fashion. Consequently, we relate Becker's construction to the construction of binary Type II codes from codes over $\mathbb{GF}(2^r)$ via the Gray map.

Acknowledgements

I would like to express my deepest gratitude to both my supervisors, Dr. Sheridan Houghten and Dr. Paul Becker, for their wonderful guidance, never-ending excitement and sense of humor throughout the course of my studies and elaboration of this work. They have always provided me with both academic and personal support, which has made an invaluable contribution to my overall experience. I would also like to recognize helpful comments and remarks of the other member of my graduate committee, Dr. Ke Qiu.

I would like to thank the whole Department of Computer Science at Brock University for creating such a friendly environment, part of which I had a chance to be. In particular, I would like to thank Dr. Brian Ross and Dr. Michael Winter for broadening my horizons within the coursework during the first year of my studies. I would like to thank Donna Phelps for her kind patience and help with the stacks of paperwork and other administrative issues, which I often supplied her with, and which I would have never been able to handle on my own. I also highly appreciate work of Cale Fairchild, who was always around to help me with any kind of technical issues, and also Baoling Bork for her positive attitude while organizing the responsibilities of the teaching assistants.

A special thank you belongs to my family and my beloved girlfriend Stephanie Zahorka for their immense support and trust. I would also like to apologize to Stephanie for making her read the entire thesis and correct my grammar mistakes.

Last but not least, I would like to thank all the other people, who kept me company during the unforgettable two years of my life—my fellow graduate students at the Department of Computer Science as well as Jan Bosák and Nina Slavíčková.

M.D.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Communication could be described as an exchange of information between two (or possibly more) communicating parties. Figure 1.1 is a simple schematic [39] of such a process. The message, produced by a sender, is encoded into a specific language and transferred through some communication channel. On the other side, it is decoded from the transmission language and passed to the destination. Ideally, the message is received by the decoder without any modifications, in the exact form as it was sent. Unfortunately, this is very difficult to guarantee in real-world applications, since hardly any communication channel is entirely free of noise or interference. In some channels, the message can be also purposely modified by an attacker who finds the communication disturbing.

The modifications in the received message are called *errors* and can have various natures. One option is that the length of the transmitted message is preserved, but the symbols are modified (substitution errors). The other types of errors are the ones that result from losing some symbols (deletion errors), introducing new ones (insertion errors), or changing the order of the symbols (transposition errors). This work concerns substitution errors only.

Errors can scarcely be eliminated, however they surely can be accounted for. For the price of transmitting additional information, we can make sure (or rather greatly increase the probability) that the communicated message will be decoded correctly. For example, the message can be transmitted several times. However, such communication would be very wasteful.

The theory of error-correcting codes, also known as coding theory, provides a mathematical aparatus for avoiding errors, which allows the communication to be both reliable and effective.

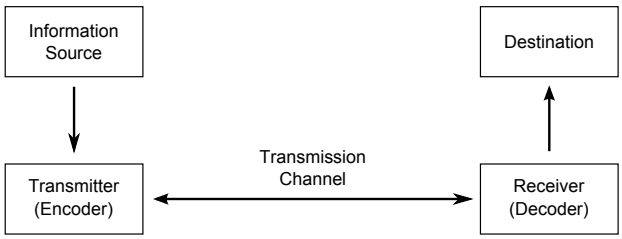

Figure 1.1: Scheme of communication as described by Shannon and Weaver [39]. The message is produced by an information source, encoded by a transmitter to a transmission language, and passed along a transmission channel. On the receiving side, the message is decoded by a receiver and passed to its destination.

Consider a very simple language of binary sequences of length 5 $\{11111, 01010, 10100\}$. These sequences, let us call them *codewords*, have a convenient property that they pairwisely differ in at least three symbols. Now imagine that in the received message, we found a word 01111. This word is most similar to the codeword 11111 as they differ in one symbol only. Thus the received word would be decoded into 11111. This principle is called *maximum likelihood decoding* and it is the basic principle of error-correction.

In fact, we can define a *sphere* of words about each of the codewords, where the *distance measure* is the number of positions in which two codewords differ. As long as the radii of the spheres remain less or equal to $\lfloor \frac{d-1}{2} \rfloor$, where d denotes the minimum distance among two codewords, we can be sure that the spheres remain disjoint. Then, every word $\mathbf{w}'$ that belongs to a sphere with center $\mathbf{w}$ will be decoded to $\mathbf{w}$ by maximum likelihood principle. The maximum sphere radius $\lfloor \frac{d-1}{2} \rfloor$ represents the *error-correcting capacity* of the code—the number of errors that the code will always be able to correct.

Some codes are more "powerful" than others. The usual requirements posed on an error-correcting code include that the code contains many codewords, can correct many errors and also that it can be quickly decoded. The first two properties are more or less dependent on the length and distance among the codewords. The requirement of fast decoding depends also on the construction of the code. Examples of classes of codes with fast decoding algorithms known are linear error-correcting codes (see Chapter 2, Section 2.4 for more details).

Error-correction is nowadays commonly present in applications where a certain level of reliability is required. It is used for data transmissions in communication networks such as radio networks, television broadcast, or even deep-space data transmissions. Error-correction is also present in both read-only and random-access memory type data storage media.

This work deals with Type II error-correcting codes, which are properly introduced in Chapter 3. In particular, it focuses on a Type II code with 2^{36} codewords of length 72 and the minimum distance 16. This code is usually referred to as a $(72, 36, 16)$ Type II code and its existence is still open. In this work, we present a methodology and exhaustive search for such a code with automorphism group $\mathbb{Z}_6$, and based on our results, we eliminate this case.

In Chapter 2, we provide the basic definitions from the field of coding theory. Chapter 3 provides an introduction into Type II codes. In particular, it summarizes the current position in the search for a $(72, 36, 16)$ Type II code with focus on its automorphism group (Section 3.1), and also gives a new construction of the extended Golay code (Section 3.2). This construction is further discussed in Chapter 5. Chapter 4 analyses the structure of a putative $(72, 36, 16)$ Type II code enforced by the automorphism group $\mathbb{Z}_6$, and in two main steps (Sections 4.2 and 4.3) shows that such a code cannot exist. In Chapter 6, we discuss how our results contribute to the search for the aforementioned code and provide directions for future work.

Chapter 2

Basic Definitions

In this chapter, we introduce the basic definitions used in the field of coding theory. We recommend [20, 38] in case of interest in further details.

2.1 Groups

Let M be a set of elements. An *n-ary operation on M* is a function $M^n \to M$, where $n \geq 0$ and M^n denotes a cartesian product of n copies of M. The number of arguments n is referred to as an *arity of an operation*. In particular, operations with arity one usually are called *unary*, and operations with arity two are called *binary*. If f is an n-ary operation on M and $f(x) \in M$ for every $x = (a_1, \ldots a_n) \in M^n$, we say that M is *closed* under f.

Let $(M, \cdot)$ be an ordered pair of set M and a binary operation $\cdot$ on M which satisfies the following:

(1) M is closed under $\cdot$ *(closure)*

(2) For all $a, b, c \in M$, $(a \cdot b) \cdot c = a \cdot (b \cdot c)$ *(associativity)*

(3) There is an element $e \in M$ such that for every $a \in M$, $a \cdot e = e \cdot a = e$. *(identity element)*

(4) For each $a \in M$, there exists an element b such that $a \cdot b = b \cdot a = e$. *(inverse elements)*

In such a case, the ordered pair $(M, \cdot)$ is called a *group*. The element e (cf. condition (3) above) is called an *identity element* and is unique. If $a \cdot b = e$

for some $a, b \in M$ and the identity element e (see (4)), element b is called an *inverse* of a, usually denoted by a^{-1}.

If $(M, \cdot)$ is a group and for every a, b in M, $a \cdot b = b \cdot a$, the group $(M, \cdot)$ is referred to as an *abelian group* and operation $\cdot$ as a *commutative operation*.

An *order of a group* $(M, \cdot)$ is the cardinality of M. A *set of generators of a group* $(M, \cdot)$ is a subset of $M' \subseteq M$ such that every element of the group can be expressed as a combination (under the group operation $\cdot$) of finitely many elements of M' and their inverses.

If $(M, \cdot)$ is a group, M' is a subset of M and $(M', \cdot)$ also forms a group, then $(M', \cdot)$ is called a *subgroup* of $(M, \cdot)$. The *trivial subgroup* of any group is the subgroup consisting of just the identity element e. A proper subgroup of a group M is a subgroup M' which is a proper subset of M', i.e. $M' \neq M$.

Two elements a, b of a group $(M, \cdot)$ are called *conjugate* if there is an element $c \in M$ such that $c \cdot a \cdot c^{-1} = b$. It can be shown that conjugacy is an equivalence relation. Therefore, it partitions the group into classes, which are usually called the *conjugacy classes*. Members of the same conjugacy class share many properties. Conjugacy is interesting especially in the case of non-abelian groups such as groups of permutations.

Let $G = (M, \cdot)$ be a group and H be a subgroup of G. Then

$$gH = \{g \cdot h \mid h \in H\}$$
$$Hg = \{h \cdot g \mid h \in H\}$$

for $g \in G$ is a *left* and *right coset* of H in G.

Examples of groups include $(\mathbb{Z}, +)$, $(\mathbb{R}, +)$, $(\mathbb{R}, \cdot)$ or $(\mathbb{Z}_n, \cdot)$ for n being a power of a prime, where $\mathbb{Z}$, $\mathbb{R}$ denote the set of integers and real numbers respectively, $\mathbb{Z}_n$ denotes the set of integers taken mod n, and operations $+, \cdot$ stand for the standard addition and multiplication resp. on those sets. The addition on the natural numbers $(\mathbb{N}, +)$ is not a group due to absence of the inverse elements.

The notion of groups is strongly connected to the notion of finite fields (see Section 2.2) and vector spaces (Section 2.3).

2.1.1 Permutation Groups

Let M be a set and $P = M^M$ the set of all projections of M to itself. An operation of *composition* $\circ$ for two projections $f, g \in P$ is defined as

$$(g \circ f)(x) = g(f(x))$$

where $x \in M$. Thus, composition is an operation on P.

Consider a set $S(M) \subseteq P$ of all the bijective projections of M to itself. Such projections are called *permutations*. For an arbitrary permutation α of set M, there is an inverse permutation α^{-1}. A composition of two inverse permutations $(\alpha \circ \alpha^{-1}) = id$ is the *identity permutation* defined as $id(x) = x$. Thus, $(S(M), \circ)$ for a set M is a group. Note that this group is not abelian.

Permutations and permutation groups deserve attention as they form the basis of the notion of automorphisms and automorphism groups (see Section 2.4.2).

Definition 2.1. *Let $i_1, i_2 \ldots, i_k$ be distinct elements of a set N. A* cycle of length k *is a permutation $\alpha \in S(N)$ such that $\alpha(i_1) = i_2, \ldots, \alpha(i_{k-1}) = i_k, \alpha(i_k) = i_1$, and α fixes all other elements of N.*

Such a permutation is denoted by $(i_1, i_2, \ldots, i_k)$. Two cycles $(i_1, i_2, \ldots, i_n)$ and $(j_1, j_2, \ldots, j_m)$ are called disjoint *if*

$$\{i_1, i_2, \ldots, i_n\} \cap \{j_1, j_2, \ldots, j_m\} = \emptyset.$$

Every permutation of a set M can be expressed as a composition of pairwise disjoint cycles. This decomposition is unique up to the order of cycles. It is a common practise to write permutations as compositions of disjoint cycles $(i_1, \ldots i_n) \circ (j_1, \ldots j_m) \circ \ldots \circ (k_1, \ldots k_q)$. Symbol $\circ$ is in this case usually omitted.

Let α be a permutation. A permutation α^m denotes a composition of m copies of α:

$$\alpha^m = \underbrace{\alpha \circ \ldots \circ \alpha}_{m}.$$

The *order of a permutation* α is the smallest positive integer k such that α^k is the identity permutation. If the permutation is expressed in disjoint cycle notation, the order is equal to the least common multiple of the lengths of those cycles. An *order of a permutation group* is the size of the group.

2.2 Finite Fields

A *field* is a set $\mathbb{F}$ with operations of addition and multiplication, commonly denoted by $+$ and $\cdot$ respectively, which satisfy the following requirements:

(1) $\mathbb{F}$ is an abelian group under $+$.

(2) If 0 denotes the identity element under $+$, the set $\mathbb{F} \setminus \{0\}$ is an abelian group under $\cdot$.

(3) Multiplication distributes over addition, i.e. $a \cdot (b+c) = (a \cdot b) + (a \cdot c)$ for $a, b, c \in \mathbb{F}$.

Field $\mathbb{F}$ is called *finite* if it has a finite number of elements. Finite fields are sometimes referred to as *Galois fields* and denoted by $\mathbb{GF}(q)$ where q is the number of elements of the field.

Proposition 2.2. *Let $\mathbb{GF}(q)$ be a finite (Galois) field with q elements. Then*

1. *$q = p^m$ for some prime p,*
2. *$p\alpha = 0$ for all $\alpha \in \mathbb{GF}(q)$,*
3. *$\mathbb{GF}(q)$ is unique up to isomorphism.*

2.3 Vector Spaces

Let $\mathbb{GF}(q)$ be a Galois field. The elements of $\mathbb{GF}(q)$ are usually called *scalars*. The *vectors* are ordered collections of scalar values.

Consider two vectors $\mathbf{a} = (a_1, a_2, \ldots a_n)$ and $\mathbf{b} = (b_1, b_2, \ldots b_n)$ with $a_i, b_i \in \mathbb{GF}(q)$, $1 \leq i \leq n$. The operations of *vector addition* $+$ and *scalar multiplication* $\cdot$ are defined as

$$\begin{aligned} \mathbf{a} + \mathbf{b} &= (a_1 + b_1, a_2 + b_2, \ldots, a_n + b_n) \\ x \cdot \mathbf{a} &= (x \cdot a_1, x \cdot a_2, \ldots, x \cdot a_n) \end{aligned}$$

where x is a scalar element from $\mathbb{GF}(q)$, and $+, \cdot$ are the operations associated with the field $\mathbb{GF}(q)$.

A *vector space* over a field $\mathbb{GF}(q)$ is a set V which is closed under vector addition and scalar multiplication. If V, V' are vector spaces and $V \subseteq V'$, we say that V is a *vector subspace* (or simply *subspace*) of V'. Every vector space V contains trivial subspaces V and $\emptyset$.

A set of vectors S from a vector space V is called *linearly independent* if there is no vector $\mathbf{v} \in S$ that can be written as a linear combination of finitely many other vectors in S.

Vector spaces are usually represented by a *basis*. A basis $B \subseteq V$ of a vector space V is a set of linearly independent vectors such that every vector $\mathbf{v} \in V$ is a linear combination of some vectors in B.

Note that a vector space can have multiple bases. One way of obtaining a new basis B' from a basis B is to replace vector $\mathbf{i}$ of B by the sum of vectors $\mathbf{i}$ and $\mathbf{j}$ for some $\mathbf{j} \neq \mathbf{i}$. In such a case, both B and B' generate the same vector space. Another possible way is to multiply vector $\mathbf{i}$ by a non-zero scalar element from $\mathbb{GF}(q)$.

For our purposes, it is convenient to view a basis as a matrix. It is often desirable to have some unique representation of a vector space. For this purpose, a basis written as a matrix in the *row reduced-echelon* form can be used. Such a matrix satisfies the following rules:

- The first non-zero entry, read from left to right, in any row is the number $1 \in \mathbb{GF}(q)$. This entry is often referred to as the *leading 1*.
- The leading one is the only non-zero entry in its column.
- The rows are ordered lexicographically, i.e. a row with leading 1 in a more-left column is always above a row with leading 1 in a more-right column.

Let V, W be subspaces of a vector space X. We define addition on vector spaces V, W as $V + W = \{\mathbf{v} + \mathbf{w} \mid \mathbf{v} \in V, \mathbf{w} \in W\}$. If $V \cap W = \{\mathbf{0}\}$, then $V + W$ is called the *direct sum* of V and W with the special notation $V \oplus W$.

2.4 Linear Codes

A *linear code* $\mathcal{C}$ of length n and dimension k is an n-dimensional vector subspace of $\mathbb{GF}(q)^n$ where $\mathbb{GF}(q)$ denotes the Galois field with q elements and q is a power of prime. Specifically, for a linear code $\mathcal{C}$, the following conditions apply:

(a) $\mathbf{u} + \mathbf{v} \in \mathcal{C}$ for all $\mathbf{u}, \mathbf{v} \in \mathcal{C}$;

(b) $a \cdot \mathbf{u} \in \mathcal{C}$ for all $\mathbf{u} \in \mathcal{C}$ and $a \in \mathbb{GF}(q)$.

The elements of a linear code are called *codewords*. A *binary code* is a code with codewords over $\mathbb{GF}(2)$. In this work, we deal with a specific kind of binary linear codes—the extremal doubly even self-dual codes.

Linear codes are usually represented by a *generator matrix*, which consists of any k linearly independent codewords. Note that a generator matrix for a code $\mathcal{C}$ of length n is a basis for the subspace of $\mathbb{GF}(q)^n$ representing the $\mathcal{C}$. Thus, one code can have many possible generator matrices.

Example 2.3. *Consider generating matrix G which is as follows:*

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

This generator matrix represents a code $\mathcal{C}$ with 16 words:

$$\begin{aligned} \mathcal{C} = \{&1000011, \quad 0000000, \quad 0110011, \quad 1101001, \\ &0100101, \quad 1100110, \quad 0101010, \quad 1011010, \\ &0010110, \quad 1010101, \quad 0011001, \quad 0111100, \\ &0001111, \quad 1001100, \quad 1110000, \quad 1111111\} \end{aligned}$$

Linear codes are usually described by attributes (n, k, d), where n is the *length* of the codewords, k is the *dimension* (rank) of the generator matrix, and d is the *minimum distance* occurring between any two distinct codewords. In this paper, we always use the Hamming distance as our distance measure.

Definition 2.4 (Hamming distance)**.** *The* Hamming distance *between words* $\mathbf{x}$ *and* $\mathbf{y}$ *is the number of positions in which they differ.*

Hence, the code presented by Example 2.3 is usually referred to as $(7, 4, 3)$ linear (Hamming) code.

For any set of k independent columns of a generator matrix G for an (n, k, d) code $\mathcal{C}$, the corresponding coordinates form the *information set* of the code $\mathcal{C}$. The remaining $(n - k)$ coordinates are called the *redundancy set* of $\mathcal{C}$. In Example 2.3, it is easy to see that we can choose the information set as columns 1, 2, 3 and 4, and the redundancy set as columns 5, 6 and 7.

The generator matrix for an (n, k, d) in the form $[I_k|A]$, where I_k denotes the $k \times k$ identity matrix, is referred to as a *generator matrix in the standard*

form. Note that a generator matrix in the standard form for a given linear code does not necessarily exist. However, for every linear code $\mathcal{C}$, there is a linear code $\mathcal{C}'$ equivalent to $\mathcal{C}$ (see Section 2.4.1) such that $\mathcal{C}'$ has a generator matrix in the standard form.

The *weight* $wt(\mathbf{w})$ of codeword $\mathbf{w}$ is the number of non-zero components in $\mathbf{w}$. The *minimum weight of a code* $\mathcal{C}$ is the minimum non-zero weight of a codeword in $\mathcal{C}$. Note that every linear code contains the all-zero codeword $\mathbf{0}$. Therefore, the minimum distance of a linear code $\mathcal{C}$ is equivalent to the minimum weight of $\mathcal{C}$.

The *weight distribution of a code* $\mathcal{C}$ is one of the important characteristics of codes. It is usually recorded as a set of pairs (i, A_i) where $i \in \mathbb{N}$ denotes a weight and A_i is the number of codewords $\mathbf{w}$ in $\mathcal{C}$ with $wt(\mathbf{w}) = i$. A code $\mathcal{C}$ is called *even* if all the codewords $\mathbf{w} \in \mathcal{C}$ have even weight. A code $\mathcal{C}$ is called *doubly even* if all the codewords $\mathbf{w} \in \mathcal{C}$ have weights divisible by 4.

A generator matrix is not the only possible compact representation of linear codes—another option is a *parity check matrix*. A Parity check matrix is an $(n-k) \times n$ matrix which consists of coefficients of $(n-k)$ parity check equations and expresses the redundancy coordinates in terms of information coordinates. See Example 2.5 for the construction.

If $G = [I_k|A]$ is a $k \times n$ generator matrix of a linear code $\mathcal{C}$ in standard form, the parity check matrix H is $H = [-A^T|I_{n-k}]$.

Example 2.5. *Consider the generator matrix for the* $(7, 4, 3)$ *Hamming code from Example 2.3:*

$$G = \begin{array}{c} \begin{matrix} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 \end{matrix} \\ \left[\begin{matrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{matrix}\right] \end{array}$$

The parity check equations put the redundancy positions u_5, u_6, u_7 *equal to the sums of information positions where the entries in* u_5, u_6, u_7 *are non-zero:*

$$\begin{aligned} u_5 &= u_2 + u_3 + u_4 \\ u_6 &= u_1 + u_3 + u_4 \\ u_7 &= u_1 + u_2 + u_4 \end{aligned}$$

The parity check equations taken mod 2 *can be written as:*

$$\begin{aligned} u_2 + u_3 + u_4 + u_5 &= 0 \\ u_1 + u_3 + u_4 + u_6 &= 0 \\ u_1 + u_2 + u_4 + u_7 &= 0 \end{aligned}$$

The resulting parity check matrix H is:

$$H = \begin{array}{c} \begin{array}{ccccccc} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 \end{array} \\ \left[\begin{array}{ccccccc} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{array}\right] \end{array}$$

Given two q-ary vectors $\mathbf{u} = (u_1, \ldots, u_n), \mathbf{v} = (v_1, \ldots, v_n)$, the *inner product* $\mathbf{u} \cdot \mathbf{v} = \Sigma_{i=1}^{n} u_i v_i \pmod{q}$. Two vectors $\mathbf{u}, \mathbf{v}$ are called *orthogonal* to each other if $\mathbf{u} \cdot \mathbf{v} = 0$. Two codes $\mathcal{C}, \mathcal{D}$ are *orthogonal* to each other if all the codewords $\mathbf{u} \in \mathcal{C}$ are orthogonal to all the codewords $\mathbf{v} \in \mathcal{D}$.

Let $\mathcal{C}$ be a code over a vector space V. We define a *dual code* $\mathcal{C}^{\perp} = \{\mathbf{u} \in V | \mathbf{u} \cdot \mathbf{v} = 0 \text{ for all } \mathbf{v} \in \mathcal{C}\}$. If $\mathcal{C}$ is an (n, k) code (saying nothing about the minimum distance), then $\mathcal{C}^{\perp}$ is an $(n, n-k)$ code. Suppose $\mathcal{C}$ has a generator matrix G and a parity check matrix H. Then, H is a generator matrix for $\mathcal{C}^{\perp}$ and G is its parity check matrix.

A code $\mathcal{C}$ is called *self-orthogonal* if all the codewords in $\mathcal{C}$ are orthogonal to each other. In such case, $\mathcal{C}$ is contained in $\mathcal{C}^{\perp}$.

Lemma 2.6 (Pless [33])**.** *All the codewords in a binary self-orthogonal $\mathcal{C}$ code must be even.*

Proof. Let $\mathbf{v} \in \mathcal{C}$ be a codeword with odd weight $wt(\mathbf{v}) = 2n + 1$, $n \in \mathbb{Z}_0$. Then $\mathbf{v} \cdot \mathbf{v} = 1$ and $\mathcal{C}$ is not self-orthogonal, a contradiction. □

The $(7, 4, 3)$ Hamming code (cf. Example 2.3) is not self-orthogonal since there are words with odd weight. The $(8, 4, 4)$ extended Hamming code, presented by Example 2.7, is self-orthogonal.

Example 2.7. *By adding a parity check coordinate to all the codewords in the generator matrix for the $(7, 4, 3)$ Hamming code (cf. Example 2.3), we obtain a generator matrix G which is as follows:*

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

This matrix is a generator matrix for the $(8, 4, 4)$ *extended Hamming code, which is self-orthogonal. This code is actually also an example of Type II code (see Chapter 3 for more details).*

Lemma 2.8 (Pless [33])**.** *Let* G *be a generator matrix for a linear binary code* $\mathcal{C}$*. Code* $\mathcal{C}$ *is self-orthogonal if and only if the rows of* G *all have even weight and are orthogonal to each other.*

Proof. The "$\Rightarrow$" direction follows from the assumption that $\mathcal{C}$ is self-orthogonal by Lemma 2.6.

For the converse ("$\Leftarrow$"), every row of G is orthogonal to itself since it has even weight. Also, any two distinct codewords from G are orthogonal to each other by the assumption. So, let us focus on the rest of the codewords.

First, focus on the even weight. Let $\mathbf{u} = \mathbf{r_i} + \mathbf{r_j}$ be a codeword obtained by a combination of two rows in G. Since both $\mathbf{r_i}$ and $\mathbf{r_j}$ have even weight and are orthogonal, word $\mathbf{u}$ has even weight too. We can repeatedly apply this construction and the argument to obtain any codeword of $\mathcal{C}$. Therefore, every codeword in $\mathcal{C}$ has even weight and is orthogonal to itself.

Let $\mathbf{u}$ and $\mathbf{v}$ be any two codewords in $\mathcal{C}$. They must be some linear combination of the rows of G. Suppose $\mathbf{u} = \mathbf{r_i} + \mathbf{r_j}$ and $\mathbf{v} = \mathbf{r_m}$. Since $\mathbf{r_i}, \mathbf{r_j}$ and $\mathbf{r_m}$ are all orthogonal to each other, we have $\mathbf{r_i} \cdot \mathbf{r_m} = 0$ and $\mathbf{r_j} \cdot \mathbf{r_m} = 0$. By axiom of finite fields,

$$(\mathbf{r_i} + \mathbf{r_j}) \cdot \mathbf{r_m} = \mathbf{r_i} \cdot \mathbf{r_m} + \mathbf{r_j} \cdot \mathbf{r_m} = 0 + 0 = 0$$

and hence, $\mathbf{u}$ is orthogonal to $\mathbf{v}$. We can repeatedly apply this argument to obtain any linear combination of the rows and at each step, the results must be orthogonal to each other. □

2.4.1 Equivalent codes

There are codes that have different sets of codewords, but they "behave" in the same way. Such codes are called *equivalent*.

Definition 2.9. *Two linear codes over* $\mathbb{GF}(q)$ *are called equivalent if one can be obtained from another by any combination of*

(a) permutation of the positions of the code; or

(b) multiplication of symbols appearing in a fixed position by a non-zero scalar $x \in \mathbb{GF}(q)$*.*

Specifically, two binary linear codes $\mathcal{C}$ and $\mathcal{D}$ are said to be *equivalent*, denoted by $\mathcal{C} \simeq \mathcal{D}$, if they can be obtained from each other by coordinate permutations (equivalently, by column permutations in one of generator matrices).

Example 2.10. *Let $\mathcal{C} = \{0000, 0011\}$, $\mathcal{D} = \{0000, 1100\}$, $\mathcal{E} = \{0000, 1010\}$ be binary linear codes and $\alpha = (1,3)(2,4)$, $\beta = (2,3)$ coordinate permutations. As $\mathcal{C} = \alpha(\mathcal{D}) = (\alpha \circ \beta)(\mathcal{E})$, the codes are pairwise equivalent.*

Equivalent codes share properties such as the minimum distance, self-orthogonality, self-duality, weight distribution and many others.

Lemma 2.11. *Let $\mathcal{R}, \mathcal{S}_1$ and $\mathcal{S}_2$ be linear codes such that $\mathcal{S}_1 \simeq \mathcal{S}_2$. The codes $\mathcal{R} \oplus \mathcal{S}_1$ and $\mathcal{R} \oplus \mathcal{S}_2$ are not necessarily equivalent.*

Proof. Consider code $\mathcal{R}$ generated by [1010] and two equivalent codes $\mathcal{S}_1 \simeq \mathcal{S}_2$ generated by [0101] and [0011] resp. Code $\mathcal{R} \oplus \mathcal{S}_1$ contains the all-one codeword $\mathbf{1}$ of weight 4 while the code $\mathcal{R} \oplus \mathcal{S}_2$ does not. As such, the codes $\mathcal{R} \oplus \mathcal{S}_1$ and $\mathcal{R} \oplus \mathcal{S}_2$ cannot be equivalent. □

A problem of equivalence of two codes can be easily reduced to a problem of graph isomorphism. An *undirected simple graph* is an ordered pair (V, E) where V is a set of vertices and E is a set of connections between two distinct vertices referred to as edges. If G is a graph, the set of vertices and edges of G are denoted by $V(G)$ and $E(G)$ respectively.

Let an undirected simple graph G model a vector space W as follows:

- Each vector $\mathbf{w} \in W$ is represented by a vertex $v \in V(G)$. For the purposes of the following, let us denote this fact by the bijection $\pi : W \to V(G)$, i.e. $\pi(\mathbf{w}) = v$.
- There is an edge $e \in E(G)$ if and only if the vectors $\mathbf{w}, \mathbf{w}'$ represented by the vertices of e have distance $d(\mathbf{w}, \mathbf{w}') = 1$.

The graph representation of the space of all binary vectors of length n is called an *n-dimensional hypercube*, denoted by Q_n.

The graph (vertex) *k-colouring of graph G* is a projection $c : V(G) \to \{0, \ldots, k-1\}$. Consider a code $\mathcal{C}$ over a vector space V that is represented

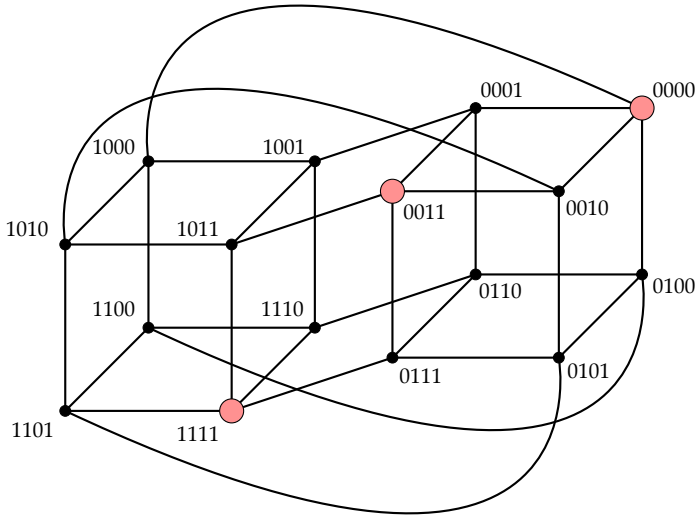

Figure 2.1: Example of the hypercubic graph representation of a binary code $\mathcal{C} = \{0000, 0011, 1111\}$.

by graph G. The code $\mathcal{C}$ is uniquely represented by a pair (G, c) where c is a 2-colouring of G formally defined as follows:

$$c(v) = \begin{cases} 1 \text{ if } \pi^{-1}(v) \in \mathcal{C} \\ 0 \text{ otherwise} \end{cases}$$

See Figure 2.1 for an example of the hypercubic representation of code $\{0000, 0011, 1111\}$.

The problem of deciding whether two codes $\mathcal{C}$ and $\mathcal{D}$ are equivalent then can be seen as a problem of deciding whether two coloured graphs (G_c, c_c) and (G_d, c_d) representing $\mathcal{C}$ and $\mathcal{D}$ respectively are isomorphic to each other, i.e. whether there is a bijection $\varphi : V(G_c) \to V(G_d)$ satisfying the following conditions:

(1) $\varphi(v)\varphi(w) \in E(G_d)$ if and only if $vw \in E(G_c)$ for all $v, w \in V(G_c)$

(2) $c_c(v) = c_d(\varphi(v))$ for all $v \in G_c$

Note that the hypercubic representation of codes is not very compact—its size depends on the size of the vector space, which is usually significantly

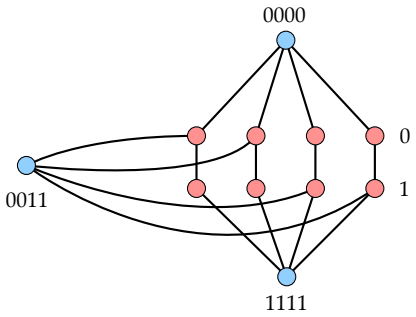

Figure 2.2: Example of the reduced graph representation of a binary code $\mathcal{C} = \{0000, 0011, 1111\}$.

larger than the code itself. Thus, it is convenient for human understanding rather than the algorithmic computations. Therefore, we provide one more, much smaller, graph representation of codes that is especially suitable for algorithmic equivalence testing (described e.g. in [19]). We refer to this representation of a code as *reduced graph representation*.

Let $\mathcal{C}$ be a q-ary code that consists of M codewords of length n. We can construct a graph G representing $\mathcal{C}$ as follows:

1. Create n *column* $q-$*tuples* of vertices denoted as $t_0 \dots t_{n-1}$. For each q-tuple t_i with $0 \leq i < n$, assign each value in $\mathbb{GF}(q)$ to a vertex in t_i and connect all the vertices in t_i to form a clique.

2. Create M vertices $v_0 \dots v_{M-1}$ to represent the codewords.

3. For each codeword $\mathbf{w_i} \in \mathcal{C}$, if value j appears on position k of the codeword i, connect the vertex v_i to the vertex with assigned value j in column q-tuple t_k.

4. Color all the vertices of column tuples with a different colour than the vertices representing the codewords.

Similarly as in the previous case, two codes $\mathcal{C}$ and $\mathcal{D}$ are equivalent if and only if their coloured (reduced) graph representations $(G_c, c_c), (G_d, c_d)$ resp. are isomorphic to each other. The size of the graph representing the code

is dependent on the size of the code, i.e. in most cases significantly smaller than the size of the whole vector space.

Note that there are other ways of representing error-correcting codes via graphs that can also be used for equivalence testing. Specifically for linear codes, we would like to point out a fast algorithm presented by [7].

Let us remark that the graph isomorphism problem is very interesting from the complexity point of view. It is known to belong to NP, however it has not been proven to be NP-hard and at the same time there is no polynomial algorithm known. However, there are systems for isomorphism testing, e.g. *nauty* (cf. [24]), which are known for their good performance. As such, they are extensively used and thus also very well tested. Our tests for code equivalence were carried out by the GAP system, package Guava [16].

2.4.2 Automorphisms

Let σ be an arbitrary column permutation and $\mathcal{C}$ a binary code. By definition, the codes $\sigma(\mathcal{C})$ and $\mathcal{C}$ are equivalent. However in some cases, the permutation σ may not change the code $\mathcal{C}$ at all, i.e. both the codes $\mathcal{C}$ and $\sigma(\mathcal{C})$ have the same codewords and $\mathcal{C} = \sigma(\mathcal{C})$. Such permutations σ are called *automorphisms*.

The set of all automorphisms of a given code $\mathcal{C}$ together with the operation of composition $\circ$ forms a (permutation) group (see Section 2.1.1 for more details on permutation groups), commonly denoted by $Aut(\mathcal{C})$. We say that code $\mathcal{C}$ *admits* an automorphism group G if G is a subgroup of $Aut(\mathcal{C})$.

Similarly to permutations, we define the *order of an automorphism* σ as the smallest positive integer k such that σ^k is the identity permutation. The *order of an automorphism group* is the size of the group.

Example 2.12. *Let $\mathcal{C}$ be the binary $(7, 4, 3)$ Hamming code (cf. Example 2.3 for a generator matrix and the set of codewords). All of the permutations $\sigma_1 = (3, 5)(6, 7)$, $\sigma_2 = (1, 3)(4, 5)$, $\sigma_3 = (2, 3)(4, 7)$, $\sigma_1 = (3, 7)(5, 6)$ are automorphisms of $\mathcal{C}$. The size of the group $Aut(\mathcal{C})$ is 168, and it is generated by $\sigma_1, \sigma_2, \sigma_3, \sigma_4$.*

Knowledge of the automorphism group and its properties can be beneficial when searching for codes, as seen for example in the backtracking algorithm presented in Section 2.4.4. The automorphism group is not only another condition that can restrict the search space, but some subcodes of the desired

code can be invariant under action of permutations in this group. Discovery of such a subcode makes a valuable starting point for an exhaustive search.

2.4.3 Operations on Codes

Many important codes arise by modifying or combining already existing codes. This section presents the usual ways of doing so. The description of the operations below was excerpted from [20, 33] and we refer to these resources for further details. All the operations on codes are presented for binary codes. Their generalization, however, is not a difficult task.

Extending

Let $\mathcal{C}$ be a linear code of length n and dimension k. Code $\mathcal{C}'$ of length $n+1$ obtained adding an overall parity check to $\mathcal{C}$ is called *extended* $\mathcal{C}$ and this operation is referred to as *extending*. Note that the extended codes always have vectors of even weight only.

Puncturing

Let $\mathcal{C}$ be a linear (n, k, d) code with a generator matrix G. By removing column i of G, we create a code $\mathcal{C}'$ of length $n-1$. Such code is called *punctured* $\mathcal{C}$ and the operation *puncturing*. The punctured code has dimension k or $k-1$ and minimum distance d or $d-1$.

Shortening

Let $\mathcal{C}$ be a linear binary (n, k, d) code $\mathcal{C}$. The *shortened* code $\mathcal{C}'$ of code $\mathcal{C}$ is the set of codewords in $\mathcal{C}$ that have entry 0 in some fixed position i, with that position deleted. If a codeword $\mathbf{w} \in \mathcal{C}$ had 1 in position i, code $\mathcal{C}'$ will be an $(n-1, k-1, d')$ code with $d' \leq d$. Otherwise, $\mathcal{C}'$ is an $(n-1, k, d)$ code.

Direct Sum

Let (n_i, k_i, d_i) be linear binary codes for $i \in \{1, 2\}$. Their *direct sum* is the $(n_1 + n_2, k_1 + k_2, \min\{d_1, d_2\})$ code

$$\mathcal{C}_1 \boxplus \mathcal{C}_2 = \{c_1 c_2 \mid c_1 \in \mathcal{C}_1, c_2 \in \mathcal{C}_2\}.$$

If $\mathcal{C}_i$ has generator matrix G_i and parity check matrix H_i, then

$$G_1 \boxplus G_2 = \begin{bmatrix} G_1 & 0 \\ 0 & G_2 \end{bmatrix} \quad \text{and} \quad H_1 \boxplus H_2 = \begin{bmatrix} H_1 & 0 \\ 0 & H_2 \end{bmatrix}$$

are a generator matrix and parity check matrix respectively for $\mathcal{C}_1 \boxplus \mathcal{C}_2$. Since the minimum distance of the direct sum of two codes does not exceed the minimum distance of either of the codes, the direct sum of two codes is generally of little use in applications and is primarily of theoretical interest.

The $(\mathbf{u}|\mathbf{u}+\mathbf{v})$ Construction

Two codes of the same length can be combined to form a third code of twice the length in a way similar to the direct sum construction. Let $\mathcal{C}_i$ be an (n, k_i, d_i) linear binary code for $i \in \{1, 2\}$. The $(\mathbf{u}|\mathbf{u}+\mathbf{v})$ *construction* produces the $(2n, k_1 + k_2, min\{2d_1, d_2\})$ code

$$\mathcal{C} = \{(\mathbf{u}|\mathbf{u}+\mathbf{v}) \mid \mathbf{u} \in \mathcal{C}_1, \mathbf{v} \in \mathcal{C}_2\}$$

.

If $\mathcal{C}_i$ has generator matrix G_i and parity check matrix H_i, then a generator matrix and parity check matrix for $\mathcal{C}$ are

$$\begin{bmatrix} G_1 & G_1 \\ 0 & G_2 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} H_1 & 0 \\ H_2 & H_2 \end{bmatrix}.$$

Unlike the direct sum construction presented above, the $(\mathbf{u}|\mathbf{u}+\mathbf{v})$ construction can produce codes that are important from a practical point of view. An example of a family with practical applications that can be constructed in this manner are the Reed-Muller codes.

2.4.4 Searching for codes

The search for a code with given parameters (n, k, d) is in general a difficult task. One of the possible approaches of tackling it (and often the best we have) is an exhaustive back-track search using the following algorithm:

```
1: function BACKTRACK(Code C)
2:     if C satisfies the required conditions then
3:         return C
4:     end if
5:     for all word w in set of candidates do
```

```
6:          C' ← C ∪ {w}
7:          if C' can satisfy the required conditions then
8:              backtrack(C')
9:          end if
10:     end for
11:     return false
12: end function
```

The set of candidates is usually formed by all the codewords of the required length, i.e. the entire $\mathbb{GF}(q)^n$ for some q, n. The conditions to be tested depend on the desired code. They generally include the obvious requirements on the minimum distance and number of codewords, but any property with a fast test available, such as linearity, self-orthogonality, self-duality and others, can provide a significant performance improvement.

However, the algorithm in this general form is hardly usable, since the search space quickly becomes enormous as the length of codewords increases. Therefore, further optimizations are necessary.

A commonly usable option is to iterate through the candidate codewords in a specific order (e.g. lexicographical). Then, at every level of recursion, only candidates $\mathbf{w}$ such that $\mathbf{w} > \mathbf{v}$ for all $\mathbf{v} \in \mathcal{C}$ need to be considered.

Usually, it is sufficient to discover one code with the required properties. Thus, traversing equivalent codes can be unnecessary. However, testing codes for inequivalence can also be a costly task. Hence, doing so at every level of recursion can be rather counter-productive. The frequency of testing that can yield the best performance is specific for every search.

All in all, the more information about the desired code that is available, the more effective this algorithm can become. In every case, the problem of searching for codes is of a hard nature, and the performance of the backtracking algorithm will always be limited.

Chapter 3

Type II Codes

A code is called *doubly even* if the weights of all the codewords are divisible by 4. Recall that for a self-orthogonal code $\mathcal{C}$, $\mathcal{C} \subseteq \mathcal{C}^{\perp}$. A *self-dual* code $\mathcal{C}$ is a code with $\mathcal{C} = \mathcal{C}^{\perp}$, i.e. $\mathcal{C} \subseteq \mathcal{C}^{\perp}$ and $\mathcal{C}^{\perp} \subseteq \mathcal{C}$.

The main interest of this work is self-dual doubly even, also referred to as *Type II*. By a theorem of Gleason, Pierce and Turyn [2], if $s > 1$ divides the weights of all the codewords in a non-trivial binary self-dual code, then $s = 2$ or $s = 4$. The weights of codewords in self-dual codes are trivially divisible by 2 (cf. Lemma 2.6). The Type II codes are in this sense even more interesting as they by definition satisfy the condition for both $s = 2$ and $s = 4$.

In the following, let $\mathbf{u} * \mathbf{v}$ be the number of common 1's in codewords $\mathbf{u}, \mathbf{v}$.

Lemma 3.1 (Pless [33])**.** *Let G be a generator matrix for a linear binary code $\mathcal{C}$. If the rows of G have weights divisible by 4 and are orthogonal to each other, then $\mathcal{C}$ is self-orthogonal and doubly even.*

Proof. By Lemma 2.8, code $\mathcal{C}$ is self-orthogonal. Suppose a codeword $\mathbf{w} \in \mathcal{C}$ is a sum of two codewords $\mathbf{u}, \mathbf{v}$ whose weight is divisible by 4. Then, the weight of $\mathbf{w}$ is

$$\begin{aligned} wt(\mathbf{w}) &= wt(\mathbf{u}) + wt(\mathbf{v}) - 2(\mathbf{u} * \mathbf{v}) \\ &= 4k + 4j - 2 \cdot 2l \\ &= 4(k + j - l). \end{aligned}$$

Since all the codewords of $\mathcal{C}$ can be generated by summing two doubly even codewords, they all have weights divisible by 4. □

Lemma 3.2 (Huffman, Pless [20]). *A code that is doubly even and has a doubly even dual code must be self-dual.*

Proof. If $\mathcal{C}$ is a doubly even code then it is self-orthogonal. If $\mathcal{C}$ and $\mathcal{C}^\perp$ are both doubly even then $\mathcal{C} \subseteq \mathcal{C}^\perp$ and $\mathcal{C}^\perp \subseteq \mathcal{C}$ so $\mathcal{C} = \mathcal{C}^\perp$. □

It has been shown (originally in [23], a very comprehensive version of the proof is available in [14]) that the minimum distance $d(\mathcal{C})$ of a Type II code $\mathcal{C}$ has upper bound

$$d(\mathcal{C}) \leq 4 + 4\lfloor \frac{n}{24} \rfloor.$$

The Type II codes with $d(\mathcal{C}) = 4 + 4\lfloor \frac{n}{24} \rfloor$ are called *extremal*.

Theorem 3.3 (Mass Formula [34]). *The number of self-dual binary codes of length n is $\Pi_{i=1}^{\frac{n}{2}-1}(2^i + 1)$.*

Proof (Excerpted from [20, page 359]). A self-dual binary code must contain the all-one vector $\mathbf{1}$. Let $\sigma_{n,k}$ denote the number of (n, k, d) self-orthogonal binary codes containing $\mathbf{1}$. First, note that $\sigma_{n,1} = 1$. Let us try to express the recurrence relation for $\sigma_{n,k}$. We illustrate the process by computing $\sigma_{n,2}$. There are $2^{n-1} - 2$ even weight vectors that are neither the all-zero vector $\mathbf{0}$ nor $\mathbf{1}$. Each of there vectors is in a unique $(n, 2, d)$ self-orthogonal code containing $\mathbf{1}$, and each such code contains two of these vectors. So, $\sigma_{n,2} = 2^{n-2} - 1$.

Every $(n, k+1)$ self-orthogonal code containing $\mathbf{1}$ contains an (n, k) self-orthogonal subcode also containing $\mathbf{1}$. Beginning with an (n, k) self-orthogonal code $\mathcal{C}$ containing $\mathbf{1}$, the only way to find an $(n, k+1)$ self-orthogonal code $\mathcal{C}'$ containing $\mathcal{C}$ is by adjoining a vector $\mathbf{c}'$ from one of the $2^{n-2k} - 1$ cosets of $\mathcal{C}$ in $\mathcal{C}^\perp$ that is unequal to $\mathcal{C}$. Thus, $\mathcal{C}$ can be extended to $2^{n-2k} - 1$ different $(n, k+1)$ self-orthogonal codes $\mathcal{C}'$. However, every such $\mathcal{C}''$ has $2^k - 1$ subcodes of dimension k containing $\mathbf{1}$. This shows that $\sigma_{n,k+1} = \frac{2^{n-2k}-1}{2^k-1}\sigma_{n,k}$. Note that using this recurrence relation, we obtain $\sigma_{n,2} = 2^{n-2} - 1$ as earlier. Using this recurrence relation, the number of self-dual binary codes of length n is

$$\begin{aligned}
\sigma_{n,n/2} &= \tfrac{2^2-1}{2^{n/2-1}-1} \cdot \tfrac{2^4-1}{2^{n/2-2}-1} \cdot \tfrac{2^6-1}{2^{n/2-3}-1} \cdots \tfrac{2^{n-2}-1}{2^1-1} \cdot \sigma_{n,1} \\
&= \tfrac{2^2-1}{2^1-1} \cdot \tfrac{2^4-1}{2^2-1} \cdot \tfrac{2^6-1}{2^3-1} \cdots \tfrac{2^{n-2}-1}{2^{n/2-1}-1} \qquad , \\
&= (2^1+1)(2^2+1)(2^3+1)\ldots(2^{n/2-1}+1)
\end{aligned}$$

which is the desired result. □

The binary Type II codes exist only for lengths divisible by 8. From the combinatorial point of view, the binary Type II codes whose length is a multiple of 24 are particularly interesting, because such codes have a strong relation to the existence of other combinatorial structures. For instance, if $\mathcal{C}$ is a Type II code of length $n = 24i$, $i \in \mathbb{N} \setminus \{0\}$, all the codewords of a fixed weight form a 5-design [1].

As we do not examine the existence of Type II codes via designs, we provide only the basic definition. We refer for instance to [22] in case of further interest in this notion.

Definition 3.4. *An incidence structure $D = (\mathcal{P}, \mathcal{B})$ is a t-(v, k, λ) design, where t, v, k, λ are non-negative integers, if $\mathcal{P}$ is a set of v elements called points and B is a collection of distinct subsets of $\mathcal{P}$ of size k called blocks such that every subset of points of size t is contained in precisely λ blocks.*

3.1 Extremal Type II Code of Length 72

The only extremal Type II codes with length divisible by 24 known so far are the extended Golay code $\mathcal{G}_{24}$ of length 24 (see Section 3.2) and the extended quadratic residue code $\mathcal{QR}_{48}$ of length 48. Both codes are unique up to equivalence (cf. Section 3.2, Lemma 3.9 for the uniqueness of $\mathcal{G}_{24}$, and [18] for the $\mathcal{QR}_{48}$).

The existence of a $(72, 36, 16)$ doubly even self-dual code is open and very important. This chapter presents the current position in the search for this code or proof of its non-existence.

The existence question for an extremal Type II code of length 72 was first proposed by N.J.A. Sloane in 1973 [40]. Sloane offered \$10 for a solution to this problem. Later, S. Dougherty offered \$100 for a proof of the existence [14] of the code, and M. Harada offered \$200 for a proof of its non-existence.

Theorem 3.5 (Rains [37])**.** *The existence of an extremal doubly even self-dual code of length $24i$, $i \in \mathbb{N} \setminus \{0\}$ is equivalent to the existence of a singly-even self-dual $(24i - 2, 12i - 1, 4i + 2)$ code.*

Corollary 3.6 is immediate:

Corollary 3.6. *A $(72, 36, 16)$ Type II code $\mathcal{C}$ exists if and only if a $(70, 35, 14)$ code exists $\mathcal{C}_{70}$.*

A_i	i
1	0, 72
249849	16, 56
18106704	20, 52
462962955	24, 48
4397342400	28, 44
16602715899	32, 40
25756721120	36

(a)

A_i	i
1	0, 70
11730	14, 56
150535	16, 54
1345960	18, 52
9393384	20, 50
49991305	22, 48
204312290	24, 46
650311200	26, 44
1627498400	28, 42
3221810284	30, 40
5066556495	32, 38
6348487600	34, 36

(b)

Table 3.1: The weight distributions of (a) a putative $(72, 36, 16)$ Type II code $\mathcal{C}$, and (b) a $(70, 35, 14)$ code $\mathcal{C}_{70}$ [14].

Code $\mathcal{C}_{70}$ referred to by Corollary 3.6 would be obtained from code $\mathcal{C}$ by puncturing two columns. The weight distribution of both the codes are known (see Table 3.1).

In this work, we focus on the search for a Type II code of length 72 using an automorphism group. The number of codes equivalent to $\mathcal{C}$ is $\frac{n!}{|Aut(\mathcal{C})|}$, thus their number decreases as the automorphism group grows. S. Dougherty in [14] suggests that the fact that any length-72 Type II code has not been discovered yet can mean two things: either the automorphism group is very large and there are very few equivalent codes to discover, or that it is very small and the internal structure of the code is not very interesting.

At this point, we can say that if such a code exists, then the latter is true as there is a long series of papers eliminating the possible orders of the automorphism group. It was shown [13] that the largest prime that may divide the order of the automorphism group of this code is 23, and this option was eliminated in [32]. In [36], 17 was eliminated. The divisor 11 is not possible either [21]. Thus, the only possible primes to divide the order of the automorphism group are 2, 3, 5 and 7.

By [28, Theorem 1], the group $Aut(\mathcal{C})$ has order 5, 7, 10, 14, or d where d divides 18 or 24, or $Aut(\mathcal{C}) \simeq A_4 \times \mathbb{Z}_3$. Yankov in [41] showed that $Aut(\mathcal{C})$ con-

tains no element of order 9, and [26, Corollary 3.6] excludes $\mathbb{Z}_{10}$ as a subgroup of $Aut(\mathcal{C})$. Finally, [15] shows that the automorphism group of a putative length-72 Type II code does not contain $\mathbb{Z}_7, \mathbb{Z}_3 \times \mathbb{Z}_3$, or D_{10} as a subgroup.

Corollary 3.7. *The automorphism group of a putative* $(72, 36, 16)$ *Type II code* $\mathcal{C}$ *has order 5 or dividing 24. Furthermore, if the order of automorphism group divides* 8*, it contains* $\mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_2$ *or the dihedral group with 8 elements* D_4 *as a subgroup.*

In addition, if $\mathcal{C}$ admits an automorphism of order 2 or 3, neither of them has any fixed points [10, 15, 20]. Let us also point out an interesting result of [29], according to which almost all the self-dual binary codes have trivial automorphism group.

3.2 Golay Codes

The extended binary Golay code is an example of a Type II code. As it is extensively referred to throughout (not only) this work, we believe that it deserves an introduction on its own.

The extended binary Golay code denoted by $\mathcal{G}_{24}$ is a $(24, 12, 8)$ linear code generated for example by the matrix G in Figure 3.2.

By puncturing the parity check column e (and in fact also any other column—Lemma 3.9) of the generator matrix, we obtain a $(23, 12, 7)$ code, which is commonly referred to as the binary Golay code[1]. The codes are named after M.J.E. Golay, who discovered them in 1949 [17].

Proposition 3.8. *The extended Golay code* $\mathcal{G}_{24}$ *has the following properties:*

(a) $\mathcal{G}_{24}$ *is doubly even*

(b) $\mathcal{G}_{24}$ *is self-orthogonal*

(c) $\mathcal{G}_{24}$ *is self-dual*

Proof. (a) and (b) hold by Lemmas 3.1 and 2.8 respectively. As the code has length $n = 24$ and dimension $\frac{n}{2} = 12$ and is self-orthogonal, it is also self-dual. □

[1]Note that there is also a pair of ternary Golay codes. As they are not significant to our problem, we do not provide any details about them. Any references to the Golay code in this work concern the extended binary Golay code $\mathcal{G}_{24}$, unless explicitly stated otherwise.

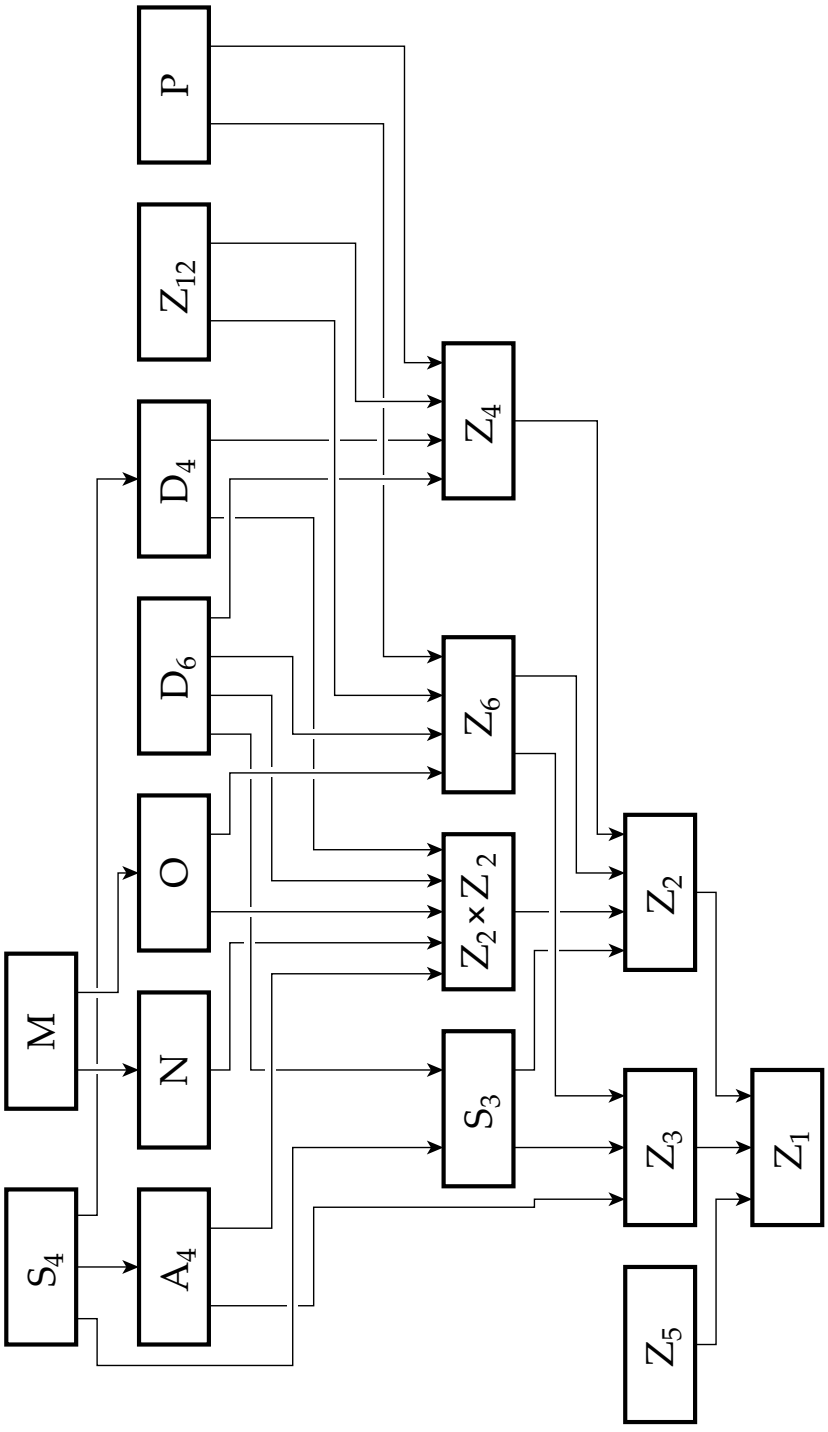

Figure 3.1: The ordering of the groups of order up to 24 which can form the automorphism group of a putative $(72, 36, 16)$ Type II code. Arrows lead always from supergroups to subgroups. The symbol M denotes the group $\mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_3$, N denotes $\mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_2$, O denotes $\mathbb{Z}_2 \times \mathbb{Z}_6$, and P denotes $\mathbb{Z}_3 \rtimes \mathbb{Z}_4$. D_4 denotes the dihedral group on four vertices, which is sometimes also referred to as D_8.

$$G = \left[\begin{array}{\overbrace{cccccccccccccccccccccc}^{\mathcal{G}_{23}}c} & e \\ 1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,0\,1\,1\,1\,0\,0\,0\,1\,0 & 1 \\ 0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,0\,1\,1\,1\,0\,0\,0\,1 & 1 \\ 0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,1\,1\,0\,1\,1\,1\,0\,0\,0 & 1 \\ 0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,1\,1\,0\,1\,1\,1\,0\,0 & 1 \\ 0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,1\,1\,0\,1\,1\,1\,0 & 1 \\ 0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,1\,1\,0\,1\,1\,1 & 1 \\ 0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,1\,0\,0\,0\,1\,0\,1\,1\,0\,1\,1 & 1 \\ 0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,1\,1\,0\,0\,0\,1\,0\,1\,1\,0\,1 & 1 \\ 0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,1\,1\,1\,0\,0\,0\,1\,0\,1\,1\,0 & 1 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,1\,1\,1\,0\,0\,0\,1\,0\,1\,1 & 1 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,1\,0\,1\,1\,1\,0\,0\,0\,1\,0\,1 & 1 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1 & 0 \end{array}\right]$$

Figure 3.2: Possible generator matrix for the extended Golay code $\mathcal{G}_{24}$ [22].

The weight distribution of $\mathcal{G}_{24}$ is as follows:

i	0	8	12	16	24
A_i	1	759	2576	759	1

Lemma 3.9 (Huffman, Pless [20])**.** *Both length* 23 *and length* 24*, possibly nonlinear, binary codes each containing the all-zero codeword* $\mathbf{0}$ *with* $M \geq 2^{12}$ *codewords and minimum distance* 7 *and* 8 *respectively are unique up to equivalence. They are the* $(23, 12, 7)$ *and* $(24, 12, 8)$ *binary Golay codes.*

A corollary of Lemma 3.9 is that by puncturing any column of a generator matrix for $\mathcal{G}_{24}$, we obtain the same code up to equivalence. Also, if we have a linear binary code $\mathcal{C}$ of length 24, it is sufficient to verify only its dimension and the minimum distance to show that this code is equivalent to the extended Golay code $\mathcal{G}_{24}$.

The *covering radius of a code* of length n is the smallest radius r such that the spheres about all the codewords with radius r contain the entire vector space $\mathbb{GF}(q)^n$. The codes with $r = \lfloor \frac{d-1}{2} \rfloor$ are called *perfect*. They are the codes capable of identifying a unique nearest neighbor for any vector in the entire vector space and as such, they are extremely effective. The binary Golay code of length 23 is a perfect code with covering radius 3. The

$(24, 12, 8)$ extended Golay code is nearly perfect as the entire space of $\mathbb{GF}(q)^{24}$ is covered by spheres with radius 4 ($= \frac{d}{2}$ for $d = 8$).

The automorphism group $Aut(\mathcal{G}_{24})$ is the Mathieu group M_{24} of order 244823040. This is rather an exceptional number among the other known Type II codes as the order of their automorphism groups tends to be rather small [29].

Up to this point, no simple way of finding Type II codes has been published. In this section, we present an interesting construction by P. Becker [3] for the $(24, 12, 8)$ extended Golay code, which is both self-dual and doubly even (see Proposition 3.8), from the $(8, 4, 4)$ extended Hamming code.

Consider the generator matrix G for the $(8, 4, 4)$ extended Hamming code $\mathcal{H}_8$ as presented in Example 2.7. We wish to obtain $(24, 12, 8)$ code, i.e. we need to accomplish the following for $\mathcal{H}_8$:

(a) Increase the length of the codewords from 8 to 24

(b) Increase the dimension of the code from 4 to 12

(c) Increase the minimum distance from 4 to 8

So, let us define a permutation $\alpha = (8, 7, 6, 5)$ and use it to create a new stacked matrix N as follows:

$$A = \begin{bmatrix} G \\ \alpha(G) \\ \alpha(G) \end{bmatrix} = \left[\begin{array}{cccccccc} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ \hline 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ \hline 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{array}\right]$$

Furthermore, let us define a projection π that maps the coordinates in the codewords of A to triples of coordinates as follows:

$$\pi(x, A) = \begin{cases} [111] \text{ if } x = [1] \text{ and } x \text{ is in the rows 1–4 of } A \\ [110] \text{ if } x = [1] \text{ and } x \text{ is in the rows 5–8 of } A \\ [011] \text{ if } x = [1] \text{ and } x \text{ is in the rows 9–12 of } A \\ [000] \text{ otherwise} \end{cases}$$

If we apply the projection π to symbols of A, we obtain the following generator matrix N:

$$N = \left[\begin{array}{cccccccc} 111 & 000 & 000 & 000 & 000 & 111 & 111 & 111 \\ 000 & 111 & 000 & 000 & 111 & 000 & 111 & 111 \\ 000 & 000 & 111 & 000 & 111 & 111 & 000 & 111 \\ 000 & 000 & 000 & 111 & 111 & 111 & 111 & 000 \\ \\ 110 & 000 & 000 & 000 & 110 & 110 & 110 & 000 \\ 000 & 110 & 000 & 000 & 000 & 110 & 110 & 110 \\ 000 & 000 & 110 & 000 & 110 & 000 & 110 & 110 \\ 000 & 000 & 000 & 110 & 110 & 110 & 000 & 110 \\ \\ 011 & 000 & 000 & 000 & 011 & 011 & 011 & 000 \\ 000 & 011 & 000 & 000 & 000 & 011 & 011 & 011 \\ 000 & 000 & 011 & 000 & 011 & 000 & 011 & 011 \\ 000 & 000 & 000 & 011 & 011 & 011 & 000 & 011 \end{array}\right]$$

Theorem 3.10. *The code generated by N is equivalent to the $(24, 12, 8)$ extended Golay code.*

We are going to prove Theorem 3.10 using Lemmas 3.11 to 3.13

Lemma 3.11. *The code generated by N is self-orthogonal.*

Proof. From Lemma 2.8, we only need to show that the rows of the generator matrix are pair-wise orthogonal. Suppose we choose a pair of rows from the first four rows. These rows were created from the generator matrix A which had pairwise orthogonal rows. The same is true for the second or third set of rows.

Note that the second and third sets of rows were both based on the matrix $\alpha(A)$. Thus, they were constructed from orthogonal rows and remain orthogonal to each other.

Now suppose we choose one row from the first set and one row from the second set. Either they have no nonzero entries in common, or their intersection consists of triples $(1, 1, 1)$ paired with triples $(1, 1, 0)$. Thus, their inner-product must be a multiple of 2, and is 0 mod 2. The rows are orthogonal. The same argument applies to a choice from the first set of rows together with a choice from the last set of rows. □

Lemma 3.12. *The dimension of the code generated by N is 12.*

Proof. Note that the original rows of the generator matrix A were linearly independent. The permuted matrix $\alpha(A)$ also has independent rows. Therefore, it is impossible to find a nontrivial linear combination solely from the first, second, or third set of rows which produces the all-zero vector.

Now suppose that there was a nontrivial sum of rows from the entire matrix which produced the all-zero vector. We could group it into a sum over the first set, second set, and third set. At least one of these partial sums is nonzero.

Now consider combinations in a single triple: no nontrivial sum of the vectors $(1, 1, 1)$, $(1, 1, 0)$, and $(0, 1, 1)$ produces $(0, 0, 0)$. Thus, the sum of rows from the entire matrix N cannot be the all-zero vector $\mathbf{0}$.

Thus, the rows of the matrix N are linearly independent, and the dimension of the generated code is 12. □

Lemma 3.13. *The minimum distance of the code generated by N is 8.*

Proof. We exhaustively verified that among all 4096 linear combinations of these rows:

- there is 1 all-zero vector;
- there are 759 vectors of weight 8;
- there are 2576 vectors of weight 12;
- there are 759 vectors of weight 16;
- and there is one vector of weight 24.

As this is a linear code, the minimum distance equals the minimum non-zero weight. □

From here, Theorem 3.10 immediately follows.

Chapter 4

Search for the (72,36,16) Type II code with automorphism group $\mathbb{Z}_6$

The main part of our effort was devoted to search for the $(72, 36, 16)$ Type II code. In particular, we focused on one of the remaining open cases for the automorphism group of this code—the group $\mathbb{Z}_6$. This chapter analyzes the structure which would be enforced by the automorphism group $\mathbb{Z}_6$. Based on this analysis, we present an algorithm and results of our exhaustive search for the aforementioned code.

4.1 Structure Related to Automorphism Group $\mathbb{Z}_6$

Suppose that a (72,36,16) Type II code exists and admits $\mathbb{Z}_6$ as an automorphism group. Such a code is in this paper referred to as $\mathcal{C}$. The assumption on automorphism group $\mathbb{Z}_6$ is equivalent to assuming that $\mathcal{C}$ admits an automorphism of the form $\zeta = \sigma \circ \theta$ where σ has order 3 and θ has order 2.

Lemma 4.1 (Huffman, Nebe, Pless)**.** *Suppose that $\mathcal{C}$ is a $(72, 36, 16)$ Type II code that admits an automorphism α. If α has order 2 or 3, then α fixes no coordinates of the code $\mathcal{C}$.*

Proof. See [15, 20]. □

With correct sorting of the columns, we can use Lemma 4.1 and assume that the desired code $\mathcal{C}$ is equivalent to a code which admits these automorphisms:

$$\begin{aligned}\sigma &= (5,3,1)(6,4,2)(11,9,7)(12,10,8)\dots(71,69,67)(72,70,68)\\ \theta &= (1,4)(2,5)(3,6)(7,10)(8,11)(9,12)\dots(67,70)(68,71)(69,72)\\ \zeta &= (1,2,3,4,5,6)(7,8,9,10,11,12)\dots(67,68,69,70,71,72)\end{aligned}$$

Each of these defines a vector space $\mathbb{F}\mathrm{ix}_\theta, \mathbb{F}\mathrm{ix}_\sigma, \mathbb{F}\mathrm{ix}_\zeta \subseteq \mathbb{GF}(2)^{72}$ consisting of all the vectors fixed by that automorphism. We can easily define a basis for each space.

4.1.1 The Space Fixed by θ

The vector space $\mathbb{F}\mathrm{ix}_\theta \subseteq \mathbb{GF}(2)^{72}$ has a basis of 36 vectors, having pattern $[100100], [010010]$ or $[001001]$ on exactly one of the sections (1,2,3,4,5,6) or (7,8,9,10,11,12), etc.

$$B_\theta = \begin{bmatrix} 100100 & 000000 & 000000 & \dots & 000000\\ 010010 & 000000 & 000000 & \dots & 000000\\ 001001 & 000000 & 000000 & \dots & 000000\\ 000000 & 100100 & 000000 & \dots & 000000\\ \dots & \dots & \dots & \dots & \dots\\ 000000 & 000000 & 000000 & \dots & 100100\\ 000000 & 000000 & 000000 & \dots & 010010\\ 000000 & 000000 & 000000 & \dots & 001001 \end{bmatrix}$$

According to [15], the code fixed by θ, denoted by $\mathbb{F}\mathrm{ix}_\theta(\mathcal{C}) = \mathcal{C} \cap \mathbb{F}\mathrm{ix}_\theta$, is isomorphic (as a vector space) to a self-dual (36,18,8) code. Up to equivalence, there are 41 such codes listed in [25] and Appendix A[1]. In this paper, we denote this family of codes by $\mathbb{L}$. In the following, we specify the correspondence between codes of $\mathbb{L}$ and codes fixed by θ in $\mathcal{C}$.

The words in $\mathbb{F}\mathrm{ix}_\theta(\mathcal{C})$ have the property (given by θ) that:

[1]The generator matrices for the 41 codes listed in [25] contain some discrepancies. As they include codewords of odd weights, the generated codes could never be self-dual. Our computations are based on the matrices listed in Appendix A that were provided by G. Nebe [27]. The author would like to thank Prof. Dr. G. Nebe for all her valuable remarks and comments.

- coordinates 1 and 4 are indistinguishable;
- coordinates 2 and 5 are indistinguishable;
- coordinates 3 and 6 are indistinguishable;
- coordinates 7 and 10 are indistinguishable etc.

Codeword $\mathbf{w}$ belongs to $\mathbb{F}\mathrm{ix}_\theta(\mathcal{C})$ if and only if $\mathbf{w} \in \mathcal{C}$ and $\mathbf{w}$ is an element of the vector space $\mathbb{F}\mathrm{ix}_\theta \subseteq \mathbb{GF}(2)^{72}$. Hence, every codeword in $\mathbb{F}\mathrm{ix}_\theta(\mathcal{C})$ can be expressed as a linear combination of the basis vectors defined above. There is a projection, in fact a homomorphism, π_θ from $\mathbb{F}\mathrm{ix}_\theta$ onto $\mathbb{GF}(2)^{36}$ that maps the consecutive segments of length 6 to segments of length 3 as follows:

$$\begin{aligned} \pi_\theta([100100]) &= [100] \\ \pi_\theta([010010]) &= [010] \\ \pi_\theta([001001]) &= [001] \\ \pi_\theta([000000]) &= [000] \end{aligned}$$

The projection maps $\mathbb{F}\mathrm{ix}_\theta(\mathcal{C})$ to a code of length 36, which is equivalent to one of the codes in $\mathbb{L}$. Conversely, we can rearrange the coordinates of $\mathcal{C}$ to produce an equivalent code $\mathcal{C}'$ such that there is a specific code $\mathcal{L} \in \mathbb{L}$, then $\pi_\theta{}^{-1}(\mathcal{L}) = \mathbb{F}\mathrm{ix}_\theta(\mathcal{C}')$.

Thus, we can state Proposition 4.2 as a consequence of action of θ:

Proposition 4.2. *If any* $(72, 36, 16)$ *Type II code that admits automorphism group* $\mathbb{Z}_6$ *exists, then there is an equivalent code* $\mathcal{C}$ *which admits automorphism* ζ*, and* $\pi_\theta^{-1}(\mathcal{L}) = \mathbb{F}ix_\theta(\mathcal{C})$ *is the subcode of* $\mathcal{C}$ *fixed by* θ *for some* $\mathcal{L} \in \mathbb{L}$*. The dimension of* $\mathcal{L}$ *is 18. Hence, 18 rows of a generator matrix for* $\mathcal{C}$ *can be computed directly from the rows of a generator matrix for some self-dual* $(36, 18, 8)$ *code* $\mathcal{L} \in \mathbb{L}$*.*

Specifically, row i *of a generator matrix for* $\mathcal{C}$ *consists of row* i *of a generator matrix for* $\mathcal{L}$ *projected by* π_θ^{-1}*, i.e. with columns* $1, 2, 3$ *repeated as columns* $1, 2, 3$ *and* $4, 5, 6$ *in* $\mathcal{C}$*; columns* $4, 5, 6$ *repeated as columns* $7, 8, 9$ *and* $10, 11, 12$ *in* $\mathcal{C}$*; etc.*

4.1.2 The Space Fixed by σ

The vector space $\mathbb{F}\text{ix}_\sigma \subseteq \mathbb{GF}(2)^{72}$ has a basis of 24 vectors having pattern [101010] or [010101] on exactly one of the sections (1,2,3,4,5,6) or (7,8,9,10,11,12), etc.

$$B_\sigma = \begin{bmatrix} 101010 & 000000 & 000000 & \dots & 000000 \\ 010101 & 000000 & 000000 & \dots & 000000 \\ 000000 & 101010 & 000000 & \dots & 000000 \\ \dots & \dots & \dots & \dots & \dots \\ 000000 & 000000 & 000000 & \dots & 101010 \\ 000000 & 000000 & 000000 & \dots & 010101 \end{bmatrix}$$

The code fixed by σ in a putative code $\mathcal{C}$ must be self-dual, doubly even, and of length 24 [20]. Thus, it must be equivalent to the extended Golay code $\mathcal{G}_{24}$ (which is, up to equivalence, the unique extremal Type II code for its length) and its generator matrix must consist of some permutation of columns in a generator matrix for $\mathcal{G}_{24}$. For brevity, we will denote this code by $\mathcal{G}$.

Similarly to π_θ, we can define homomorphism $\pi_\sigma : \mathbb{F}\text{ix}_\sigma \to \mathbb{GF}(2)^{24}$:

$$\begin{aligned} \pi_\sigma([101010]) &= [10] \\ \pi_\sigma([010101]) &= [01] \\ \pi_\sigma([000000]) &= [00] \end{aligned}$$

This projection maps the fixed code $\mathbb{F}\text{ix}_\sigma(\mathcal{C}) = \mathbb{F}\text{ix}_\sigma \cap \mathcal{C}$ to $\mathcal{G}$ and is undefined for the rest of the codewords in $\mathcal{C}$. Thus, if $\mathcal{G} \simeq \mathcal{G}_{24}$, code $\mathbb{F}\text{ix}_\sigma(\mathcal{C})$ can be constructed as $\pi_\sigma^{-1}(\mathcal{G})$.

Proposition 4.3. *Suppose $\mathcal{C}$ is a $(72, 36, 16)$ Type II code that admits automorphism ζ, and $\pi_\sigma^{-1}(\mathcal{G}) = \mathbb{F}ix_\sigma(\mathcal{C})$ is a subcode of $\mathcal{C}$ fixed by σ. The dimension of $\mathcal{G}$ is 12. Thus, 12 rows of a generator matrix for $\mathcal{C}$ can be computed from the rows of a generating matrix for $\mathcal{G} \simeq \mathcal{G}_{24}$.*

There is a 6 dimensional overlap of the code fixed by automorphism σ and θ in $\mathcal{C}$ (proof of this claim is provided by Lemmas 4.6 and 4.14). Thus, row $j = i + 12$ of a generator matrix for $\mathcal{C}$ consists of row i of a generator matrix for $\mathcal{G}$ projected by π_σ^{-1}, i.e. with columns $1, 2$ repeated as columns $1, 3, 5$ and $2, 4, 6$ in $\mathcal{C}$; columns $3, 4$ repeated as columns $7, 9, 11$ and $8, 10, 12$ in $\mathcal{C}$; etc.

We will show in Section 4.1.3 that $\mathbb{F}\mathrm{ix}_\theta(\mathcal{C}) \cap \mathbb{F}\mathrm{ix}_\sigma(\mathcal{C})$ has dimension at least 6 (see Lemma 4.6) and later in Section 4.2.2 that the dimension actually is precisely 6 (Lemma 4.14). Combining this fact with Propositions 4.2 and 4.3, we can conclude that if a $(72, 36, 16)$ Type II code exists, then there is an equivalent code $\mathcal{C}$ which admits a generator matrix with the following properties: The first 18 rows are fully determined by the rows of a generating matrix for some $\mathcal{L} \in \mathbb{L}$. The next 6 rows are fully determined by the rows of a generating matrix for some $\mathcal{G} \simeq \mathcal{G}_{24}$. Thus in total, 24 of 36 rows for a generating matrix can be determined.

4.1.3 The Space Fixed by ζ

The vector space $\mathbb{F}\mathrm{ix}_\zeta \subseteq \mathbb{GF}(2)^{72}$ has a basis of 12 vectors having pattern $[1, 1, 1, 1, 1, 1]$ on exactly one of the sections (1,2,3,4,5,6) or (7,8,9,10,11,12), etc. Note that $\mathbb{F}\mathrm{ix}_\zeta$ is a subspace of each of the larger spaces $\mathbb{F}\mathrm{ix}_\sigma$ and $\mathbb{F}\mathrm{ix}_\theta$, i.e. $\mathbb{F}\mathrm{ix}_\zeta = (\mathbb{F}\mathrm{ix}_\sigma \cap \mathbb{F}\mathrm{ix}_\theta)$.

$$B_\zeta = \begin{bmatrix} 111111 & 000000 & 000000 & \dots & 000000 \\ 000000 & 111111 & 000000 & \dots & 000000 \\ \dots & \dots & \dots & \dots & \dots \\ 000000 & 000000 & 000000 & \dots & 111111 \end{bmatrix}$$

We would like to show that the dimension of code fixed in $\mathcal{C}$ by automorphism ζ is at least 6. For this purpose, we use a technique introduced by [9]. Let the code fixed in $\mathcal{C}$ by ζ be denoted by $\mathbb{F}\mathrm{ix}_\zeta(\mathcal{C}) = \mathbb{F}\mathrm{ix}_\theta(\mathcal{C}) \cap \mathbb{F}\mathrm{ix}_\sigma(\mathcal{C})$. In addition, let us define projections $\pi_\zeta : \mathbb{GF}(2)^{72} \to \mathbb{GF}(2)^{12}$ and $\phi : \mathbb{GF}(2)^{72} \to \mathbb{GF}(2)^{12}$ as

$$\begin{aligned} \pi_\zeta([111111]) &= [1] \\ \pi_\zeta([000000]) &= [0] \end{aligned}$$

and

$$\phi((c_1, c_2, \dots, c_{72})) = (\Sigma_{i=1}^{6} c_i, \Sigma_{i=7}^{12} c_i, \dots, \Sigma_{i=67}^{72} c_i).$$

Projection π_ζ maps the consecutive 6-tuples of coordinates to the first coordinate, and projection ϕ sums the six-cycles (mod 2).

Lemma 4.4. *Code $\phi(\mathcal{C})$ is self-orthogonal.*

Proof. Choose any $\mathbf{v} = (v_1, v_2, \ldots, v_{72})$ and $\mathbf{w} = (w_1, w_2, \ldots, w_{72})$ in code $\mathcal{C}$. Then $\phi(\mathbf{v})$ and $\phi(\mathbf{w})$ are in code $\phi(\mathcal{C})$. Consider the inner product of $\phi(\mathbf{v})$ and $\phi(\mathbf{w})$:

$$\begin{aligned}
\phi(\mathbf{v}) \cdot \phi(\mathbf{w}) &= \Sigma_{i=1}^{6} v_i \cdot \Sigma_{i=1}^{6} w_i + \cdots + \Sigma_{i=67}^{72} v_i \cdot \Sigma_{i=67}^{72} w_i \\
&= (w_1 \cdot \Sigma_{i=1}^{6} v_i) + (w_2 \cdot \Sigma_{i=1}^{6} v_i) + \ldots + (w_{72} \cdot \Sigma_{i=67}^{72} v_i) \\
&= (\mathbf{v} + \zeta(\mathbf{v}) + \zeta^2(\mathbf{v}) + \zeta^3(\mathbf{v}) + \zeta^4(\mathbf{v}) + \zeta^5(\mathbf{v})) \cdot \mathbf{w} \\
&= 0 + 0 + 0 + 0 + 0 + 0 = 0
\end{aligned}$$

Notice that $\zeta(\mathbf{v})$, etc., are all in $\mathcal{C}$, which is self-dual, so their inner products with $\mathbf{w}$ are zeroes. Thus, $\phi(\mathcal{C})$ is self-orthogonal. □

Lemma 4.5. *Code $\pi_\zeta(\mathbb{F}ix_\zeta(\mathcal{C}))$ is the dual code of $\phi(\mathcal{C})$.*

Proof. Choose an $\mathbf{f} = (f_1, f_2, \ldots, f_{72})$ in $\pi_\zeta(\mathbb{F}\mathrm{ix}_\zeta(\mathcal{C}))$ and any $\mathbf{v}$ in $\mathcal{C}$. Note that $f_1 = f_2 = \ldots = f_6$, $f_7 = f_8 = \ldots = f_{12}$, etc.

$$\begin{aligned}
\pi_\zeta(\mathbf{f}) \cdot \phi(\mathbf{v}) &= f_1 \cdot (v_1 + \cdots + v_6) + f_7 \cdot (v_7 + \cdots + v_{12}) + \ldots \\
&= f_1 v_1 + f_1 v_2 + f_1 v_3 + \ldots \\
&= f_1 v_1 + f_2 v_2 + f_3 v_3 + \ldots \\
&= \mathbf{f} \cdot \mathbf{v} \\
&= 0
\end{aligned}$$

Thus, $\mathbf{f}$ is an element of the dual code $[\phi(\mathcal{C})]^\perp$. In general, we have $\pi_\zeta(\mathbb{F}\mathrm{ix}_\zeta(\mathcal{C})) \subseteq [\phi(\mathcal{C})]^\perp$.

Now choose any element $\mathbf{g} \in [\phi(\mathcal{C})]^\perp$. Note that this is a vector of length 12. Choose any $\mathbf{r}' \in \phi(\mathcal{C})$. For any preimage $\mathbf{r} \in \mathcal{C}$ of $\mathbf{r}'$ holds that

$$\begin{aligned}
\mathbf{g} \cdot \mathbf{r}' &= g_1 r'_1 + g_2 r'_2 + \cdots + g_{12} r'_{12} \\
&= g_1(r_1 + r_2 + \cdots + r_6) + g_2(r_7 + r_8 + \cdots + r_{12}) + \ldots
\end{aligned}$$

Now if we define $\mathbf{f} = \pi_\zeta^{-1}(\mathbf{g}) = (g_1, g_1, g_1, g_1, g_1, g_1, g_2, g_2, \dots)$, then this is $\mathbf{g} \cdot \mathbf{r}' = \mathbf{f} \cdot \mathbf{r}$. Since $\mathbf{g}$ is in $[\phi(\mathcal{C})]^\perp$, $\mathbf{g} \cdot \mathbf{r}' = 0$. Thus, $\mathbf{f} \cdot \mathbf{r} = 0$ for every $\mathbf{r} \in \mathcal{C}$. Since $\mathcal{C}$ is self-dual, this means $\mathbf{f} \in \mathcal{C}$ and therefore, $\mathbf{f} \in \mathbb{F}\mathrm{ix}_\zeta(\mathcal{C})$. We can conclude that $\mathbf{g} \in [\phi(\mathcal{C})]^\perp$ implies $\mathbf{g} \in \pi_\zeta(\mathbb{F}\mathrm{ix}_\zeta(\mathcal{C}))$. In general, $[\phi(\mathcal{C})]^\perp \subseteq \pi_\zeta(\mathbb{F}\mathrm{ix}_\zeta(\mathcal{C}))$. □

Lemma 4.6. *The code $\mathbb{F}ix_\zeta(\mathcal{C})$ fixed in $\mathcal{C}$ by ζ has dimension at least 6.*

Proof. Note that $\pi_\zeta(\mathbb{F}\mathrm{ix}_\zeta(\mathcal{C}))$ has the same dimension as $\mathbb{F}\mathrm{ix}_\zeta(\mathcal{C})$ as π_ζ only discards the redundant repetitions of every coordinate. Thus showing that the dimension of $\pi_\zeta(\mathbb{F}\mathrm{ix}_\zeta(\mathcal{C}))$ is at least 6 is equivalent to the desired claim.

Code $\phi(\mathcal{C})$ is a self-orthogonal code of length 12 by Lemma 4.4. Thus, its dimension cannot be more than $k = \frac{12}{2} = 6$. By Lemma 4.5, the code $\pi_\zeta(\mathbb{F}\mathrm{ix}_\zeta(\mathcal{C}))$ is the dual code to $\phi(\mathcal{C})$. Its dimension must be $12 - k$. Thus, $\mathbb{F}\mathrm{ix}_\zeta(\mathcal{C})$ has dimension at least 6. □

Projection π_ζ maps subcode $\mathbb{F}\mathrm{ix}_\zeta(\mathcal{C}) \subseteq \mathcal{C}$ onto a code of length 12 that we denote by $\mathcal{D}$. In Section 4.2.2, we show that this code is unique up to equivalence, and its dimension is precisely 6.

4.2 Step 1: Search for Golay Codes

From the previous sections, we can deduce the following:

Proposition 4.7. *Suppose that there is some $(72, 36, 16)$ Type II code $\mathcal{C}'$ that admits $\mathbb{Z}_6$ automorphism group. Then, there is an equivalent $(72, 36, 16)$ Type II code $\mathcal{C}$ which can be expressed as $\mathcal{C} = \pi_\theta^{-1}(\mathcal{L}) + \pi_\sigma^{-1}(\mathcal{G}) \oplus \mathcal{Q}$ where $\mathcal{L}$ is a self-dual $(36, 18, 8)$ code which is fixed under an automorphism of order 3 with no fixed points, $\mathcal{G}$ is equivalent to the extended Golay code $\mathcal{G}_{24}$, and $\mathcal{Q}$ is so far undetermined.*

Proof. We have shown that any $(72, 36, 16)$ Type II code with automorphism group $\mathbb{Z}_6$ can be expressed as a sum of some code $\mathcal{L}'$ equivalent to an $\mathcal{L} \in \mathbb{L}$, and some Golay code $\mathcal{G} \simeq \mathcal{G}_{24}$, and an unknown component $\mathcal{Q}$ of dimension 12. The code can be expressed as $(\pi_\theta^{-1}(\mathcal{L}') + \pi_\sigma^{-1}(G')) \oplus \mathcal{Q}'$

Note that $\pi_\theta^{-1}(\mathcal{L}')$ is the subcode of $\mathcal{C}'$ contained in the vector space $\mathbb{F}\mathrm{ix}_\theta$. Now σ is an automorphism of the $\mathcal{C}'$, so $\mathbb{F}\mathrm{ix}_\theta(\mathcal{C}') = \mathbb{F}\mathrm{ix}_\theta(\sigma(\mathcal{C}'))$. Examining our basis for $\mathbb{F}\mathrm{ix}_\theta$, we see that σ simply permutes the basis elements. Thus,

any element of $\mathbb{F}\mathrm{ix}_\theta(\mathcal{C}')$ is mapped back into $\mathbb{F}\mathrm{ix}_\theta(\mathcal{C}')$. This subcode is preserved by σ. Since $\mathcal{L}'$ is the image of $\mathbb{F}\mathrm{ix}_\theta(\mathcal{C}')$ under the projection π_θ, it must admit an automorphism of order 3 with no fixed points. Specifically, it must admit $\pi_\theta(\sigma(\pi_\theta^{-1}))$. We know that $\mathcal{L}'$ is equivalent to some $\mathcal{L} \in \mathbb{L}$. Permuting columns to transform $\mathcal{L}'$ into $\mathcal{L}$, we obtain new second and third components $\mathcal{G} \simeq \mathcal{G}_{24}$ and $\mathcal{Q}$. $\mathcal{L}$ is still preserved by an automorphism of order 3 with no fixed points, though it may not be identical to σ.

□

Corollary 4.8. *If a* $(72, 36, 16)$ *Type II code that admits* $\mathbb{Z}_6$ *automorphism group exists, then there is an equivalent code* $\mathcal{C}$ *such that* $\mathbb{F}ix_\theta(\mathcal{C})$ *is one of thirteen* $(36, 18, 8)$ *self-dual codes which admit fixed-point-free automorphism of order 3.*

Proof. Calculations by G. Nebe [27]. □

Note that even though only 13 out of 41 codes in $\mathbb{L}$ admit an automorphism without any fixed points, there are in total 19 cases of a code $\mathcal{L} \in \mathbb{L}$ and such automorphism (up to conjugacy). See Section 4.2.5 for further details.

Our approach essentially first attempts to build a putative code $\mathcal{C}$ as a sum of overlapping vector spaces $\pi_\theta^{-1}(\mathcal{L}) + \pi_\sigma^{-1}(\mathcal{G}) \oplus \mathcal{Q}$ where $\mathcal{L} \in \mathbb{L}$ and $\mathcal{G} \simeq \mathcal{G}_{24}$ are the subcodes fixed by θ and σ respectively (cf. Sections 4.1.1 and 4.1.2), and $\mathcal{Q}$ is the remaining subcode of $\mathcal{C}$ of dimension 12. We attempt to compose $\mathcal{C}$ precisely in this order, i.e. by fixing $\pi_\theta^{-1}(\mathcal{L})$ by choosing a code $\mathcal{L} \in \mathbb{L}$, adding component $\pi_\sigma^{-1}(\mathcal{G})$ to it and by filling $\mathcal{Q}$ at last. In this section, we describe an exhaustive search for $(72, 24, d = 16)$ codes that can be built from codes of $\mathbb{L}$ and $\mathcal{G} \simeq \mathcal{G}_{24}$.

Let us first emphasize the main complication of this phase of the search. Our task is to construct all the non-equivalent codes $(\pi_\theta^{-1}(\mathcal{L}) + \pi_\sigma^{-1}(\mathcal{G}))$. For a fixed code $\mathcal{L}$, which is one of the 41 candidates in $\mathbb{L}$, we must consider all the possible 24! codes $\mathcal{G}$ (permutations of $\mathcal{G}_{24}$) as they can create non-equivalent codes (recall Lemma 2.11). To reduce the search space, let us focus on what restrictions on code $\mathcal{G} \simeq \mathcal{G}_{24}$ are posed by fixing $\mathcal{L} \in \mathbb{L}$.

4.2.1 Weight Restrictions

Assume that code $\mathcal{L}$ and its generator matrix G are both fixed. We are looking for all the codes $\mathcal{G} \simeq \mathcal{G}_{24}$ such that code $(\pi_\theta^{-1}(\mathcal{L}) + \pi_\sigma^{-1}(\mathcal{G}))$ can form

a subcode of $\mathcal{C}$. As $\mathcal{C}$ is a Type II code with minimum distance 16, all the codewords $\mathbf{w} \in (\pi_\theta^{-1}(\mathcal{L}) + \pi_\sigma^{-1}(\mathcal{G})) \subseteq \mathcal{C}$ must meet the weight restriction $wt(\mathbf{w}) \in \mathfrak{W}$, where $\mathfrak{W} = \{0, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 56, 72\}$.

To indicate that composition of two codes does not violate restrictions given by weight distribution of code $\mathcal{C}$, we introduce the notion of *compatibility*. We say that a codeword $\mathbf{w}$ is *compatible* with code $\mathcal{R}$ for $\mathcal{C}$ if $wt(\mathbf{w} + \mathbf{u}) \in \mathfrak{W}$ for every $\mathbf{u} \in \mathcal{R}$. Code $\mathcal{S}$ is *compatible* with $\mathcal{R}$ if all the codewords of $\mathcal{S}$ are compatible with $\mathcal{R}$. In particular for $\mathcal{L}$ and $\mathcal{G}$—codes of lengths 36 and 24 respectively—we say that $\mathcal{G}$ is *compatible* with code $\mathcal{L}$ if $\pi_\sigma^{-1}(\mathcal{G})$ is compatible with $\pi_\theta^{-1}(\mathcal{L})$.

Presumably, not all the codes $\mathcal{G} \simeq \mathcal{G}_{24}$ are compatible with all $\mathcal{L} \in \mathbb{L}$. For a fixed code $\mathcal{L} \in \mathbb{L}$, the only interesting codes $\mathcal{G}$ are the ones with $\mathbf{w} \in \mathfrak{R}$ for every $\mathbf{w} \in \mathcal{G}$, where $\mathfrak{R} \subseteq \mathbb{GF}(2)^{24}$ is the set of all words compatible with code $\mathcal{L}$.

There is a significant shortcut here if we notice that the basis vectors of both $\mathbb{F}\mathrm{ix}_\theta$ and $\mathbb{F}\mathrm{ix}_\sigma$ have a pattern on segments of length 6 corresponding to six-cycles of ζ. This pattern can be used to determine the weight of a codeword $\mathbf{w} = \pi_\theta^{-1}(\mathbf{i}) + \pi_\sigma^{-1}(\mathbf{g})$ in $\pi_\theta^{-1}(\mathcal{L}) + \pi_\sigma^{-1}(\mathcal{G})$ where $\mathbf{i} \in \mathcal{L}, \mathbf{g} \in \mathcal{G}$.

The pattern of the component $\pi_\theta^{-1}(\mathbf{i})$ on a six-cycle is determined by three coordinates of $\mathbf{i}$. Similarly, the pattern of $\pi_\sigma^{-1}(\mathbf{g})$ on a six-cycle is determined by two coordinates of $\mathbf{g}$. Table 4.1 summarizes the weights of all the possible patterns $(\pi_\theta^{-1}(\mathbf{i}) + \pi_\sigma^{-1}(\mathbf{g}))$. We can observe that the weights $(\pi_\theta^{-1}(\mathbf{i}) + \pi_\sigma^{-1}(\mathbf{g}))$ are invariant if the weights of three- and two-coordinate segments of $\mathbf{i}$ and $\mathbf{g}$ respectively are also fixed.

Corollary 4.9. *The weight of $(\pi_\theta^{-1}(\mathbf{i}) + \pi_\sigma^{-1}(\mathbf{g}))$ can be determined solely based on the weights of $\mathbf{i} \in \mathcal{L}$ and $\mathbf{g} \in \mathcal{G}$ on triples and pairs of coordinates respectively.*

We now present two examples to illustrate how this weight can be determined and how a codeword from $\mathcal{L}$ can eliminate a candidate for a codeword in $\mathcal{G}$.

Example 4.10. *Consider words $\mathbf{i} \in \mathcal{L}$ and $\mathbf{g} \in \mathcal{G}$:*

$$\begin{aligned} \mathbf{i} &= 100\ 000\ 000\ 000\ 000\ 000\ 001\ 000\ 001\ 101\ 100\ 011 \\ \mathbf{g} &= 11\ 10\ 00\ 00\ 00\ 00\ 00\ 01\ 11\ 11\ 11\ 11 \end{aligned}$$

The weights of the word on 3-segments for $\mathbf{i}$ and 2-segments for $\mathbf{g}$ form

	$\mathcal{G}$				
$\mathcal{L}$	00	01	10	11	$\pi_\theta^{-1}(\mathcal{L})$
000	0	3	3	6	000000
001	2	3	3	4	001001
010	2	3	3	4	010010
011	4	3	3	2	011011
100	2	3	3	4	100100
101	4	3	3	2	101101
110	4	3	3	2	110110
111	6	3	3	0	111111
	000000	010101	101010	111111	
	$\pi_\sigma^{-1}(\mathcal{G})$				

Table 4.1: Possible weights of codeword $(\pi_\theta^{-1}(\mathbf{i}) + \pi_\sigma^{-1}(\mathbf{g}))$ for $\mathbf{i} \in \mathcal{L}$, $\mathbf{g} \in \mathcal{G}$ on segments of length 6.

the following two vectors:

$$\begin{aligned} \mathbf{i}^* &= (1,0,0,0,0,0,1,0,1,2,1,2) \\ \mathbf{g}^* &= (2,1,0,0,0,0,0,1,2,2,2,2) \end{aligned}$$

Assume we extend $\mathbf{i}$ *and* $\mathbf{g}$ *to length* 72 *as described in Sections 4.1.1 and 4.1.2. Then, use Table 4.1 to determine the weight of each* 6*-cycle of* $\pi_\theta^{-1}(\mathbf{i}) + \pi_\sigma^{-1}(\mathbf{g})$*. The vector of those weights* $\mathbf{w}^*$ *is:*

$$\mathbf{w}^* = (4,3,0,0,0,0,2,3,4,2,4,2)$$

The total weight of word $\mathbf{w}$ *(which has length* 72*) obtained by summing up the entries of the vector* $\mathbf{w}^*$ *is* 24*. Hence, the codeword* $\mathbf{w}$ *does not violate the rules given by weights and may possibly belong to an extremal Type II code* $\mathcal{C}$ *of length* 72*.*

Example 4.11. *Consider words* $\mathbf{i} \in \mathcal{L}$ *and* $\mathbf{g} \in \mathcal{G}$*:*

$$\begin{aligned} \mathbf{i} &= 101\ 100\ 000\ 000\ 100\ 000\ 000\ 000\ 000\ 000\ 101\ 101 \\ \mathbf{g} &= 10\ 10\ 10\ 10\ 00\ 00\ 00\ 00\ 00\ 00\ 11\ 11 \end{aligned}$$

The word $\mathbf{w} = \pi_\theta^{-1}(\mathbf{i}) + \pi_\sigma^{-1}(\mathbf{g})$ *of length* 72 *has weight vector:*

$$\mathbf{w}^* = (3,3,3,3,2,0,0,0,0,0,2,2)$$

and its total weight is 18. *As the combined word does not have doubly even weight, the codes* $\mathcal{G}$ *and* $\mathcal{L}$ *are not compatible for length-*72 *extremal Type II code* $\mathcal{C}$.

We use Corollary 4.9 for effective numeration of a set of words that possibly can form code $\mathcal{G}$ for some fixed code $\mathcal{L}$. We denote this set of candidates by $\mathfrak{R}$.

To formalize the notion and describe the algorithm precisely, let us define a projection $\omega : \mathbb{GF}(2)^{24} \to \{0,1,2\}^{12}$, creating a vector consisting of the weights of the codewords on pairs of coordinates. Furthermore, let $\sim$ be a relation on $\mathbb{GF}(2)^{24}$ defined by:

$$\mathbf{x} \sim \mathbf{y} \Leftrightarrow \omega(\mathbf{x}) = \omega(\mathbf{y})$$

Relation $\sim$ is an equivalence on $\mathbb{GF}(2)^{24}$. For each class of equivalence $C \in \mathbb{GF}(2)^{24}/\sim$, either all the vectors $\mathbf{w} \in C$ are compatible with code $\mathcal{L}$, or none of them is. By Corollary 4.9, the compatibility of a class that contains $\mathbf{w}$ can be determined via its unique representation $\omega(\mathbf{w})$. This provides clear directions for the following algorithm for finding set $\mathfrak{R}$:

Step 1: Exhaustively find set $\mathfrak{V}$ of all vectors $\mathbf{v} \in \{0,1,2\}^{12}$ that represent classes of equivalence of codes $\mathcal{G}$ compatible with code $\mathcal{L}$

Step 2: Determine $\mathfrak{R}$ as the union of classes in $\mathfrak{V}$

4.2.2 Intersection of $\pi_\theta^{-1}(\mathcal{L})$ and $\pi_\sigma^{-1}(\mathcal{G})$

The overlap $\mathbb{F}\mathrm{ix}_\zeta(\mathcal{C})$ between $\mathbb{F}\mathrm{ix}_\theta(\mathcal{C})$ and $\mathbb{F}\mathrm{ix}_\sigma(\mathcal{C})$ of dimension at least 6 (see Section 4.1.3) must be reflected in codes $\mathcal{L}$ and $\mathcal{G}$. Let us denote the subcodes that extend to $\mathbb{F}\mathrm{ix}_\zeta(\mathcal{C})$ from $\mathcal{G}$ and $\mathcal{L}$ by $\mathcal{F} \subseteq \mathcal{G}$ and $\mathcal{E} \subseteq \mathcal{L}$.

Lemma 4.12. *Code* $\mathcal{F} \subseteq \mathcal{G}$ *of dimension at least 6 can be directly determined from* $\mathcal{L}$.

Proof. All the vectors in $\mathbb{F}\mathrm{ix}_\zeta(\mathcal{C}) \subseteq \mathbb{F}\mathrm{ix}_\zeta$ have all the segments of length 6 corresponding to 6-cycles of ζ of weight 0 or 6 (patterns [000000] or [111111]). Hence, both the pairs of coordinates in code $\mathcal{F} \subseteq \mathcal{G}$ and triples of coordinates

in $\mathcal{E} \subseteq \mathcal{L}$ must be indistinguishable. This allows us to easily compute $\mathcal{E}$ from the code $\mathcal{L}$. By replacing triples of coordinates by one in $\mathcal{E}$, we obtain code $\mathcal{D}$ of length 12 (cf. Section 4.1.3). Consequently, code $\mathcal{F} \subseteq \mathcal{G}$ is obtained from $\mathcal{D}$ by replacing every coordinate with two copies of it. □

Example 4.13. *Consider generator matrix $G_{\mathcal{L}}$ for $\mathcal{L}$ which is as follows:*

$$G_{\mathcal{L}} = \left[\begin{array}{cccccccccccc}
001 & 000 & 011 & 010 & 010 & 010 & 000 & 000 & 011 & 100 & 010 & 000 \\
000 & 001 & 001 & 001 & 010 & 000 & 000 & 001 & 001 & 000 & 011 & 101 \\
000 & 001 & 011 & 000 & 000 & 010 & 000 & 010 & 000 & 000 & 111 & 000 \\
001 & 001 & 100 & 000 & 000 & 000 & 000 & 000 & 000 & 000 & 111 & 011 \\
000 & 001 & 011 & 000 & 000 & 000 & 000 & 000 & 111 & 001 & 001 & 000 \\
000 & 001 & 011 & 000 & 000 & 001 & 000 & 001 & 000 & 101 & 010 & 000 \\
001 & 100 & 010 & 000 & 000 & 000 & 000 & 000 & 000 & 101 & 010 & 011 \\
001 & 000 & 000 & 000 & 010 & 010 & 000 & 101 & 010 & 101 & 010 & 100 \\
101 & 000 & 000 & 000 & 000 & 000 & 000 & 000 & 000 & 101 & 101 & 110 \\
001 & 000 & 001 & 100 & 000 & 010 & 000 & 001 & 010 & 010 & 000 & 010 \\
001 & 001 & 000 & 000 & 000 & 010 & 010 & 000 & 001 & 110 & 000 & 111 \\
001 & 001 & 011 & 000 & 010 & 110 & 000 & 001 & 010 & 110 & 110 & 100 \\
001 & 001 & 010 & 000 & 010 & 010 & 100 & 001 & 000 & 011 & 001 & 101 \\
000 & 000 & 000 & 000 & 011 & 000 & 000 & 000 & 011 & 011 & 011 & 000 \\
011 & 000 & 000 & 000 & 000 & 000 & 000 & 000 & 000 & 011 & 011 & 101 \\
000 & 000 & 010 & 000 & 000 & 000 & 001 & 001 & 011 & 111 & 100 & 010 \\
001 & 010 & 001 & 000 & 000 & 000 & 000 & 000 & 000 & 011 & 100 & 011 \\
000 & 001 & 011 & 000 & 110 & 000 & 000 & 000 & 001 & 111 & 111 & 000
\end{array}\right]$$

This code $\mathcal{L}$ contains a subcode $\mathcal{E}$ with generator matrix

$$G_{\mathcal{E}} = \left[\begin{array}{cccccccccccc}
111 & 000 & 111 & 111 & 000 & 000 & 000 & 000 & 000 & 000 & 000 & 111 \\
111 & 111 & 111 & 000 & 000 & 000 & 000 & 000 & 000 & 000 & 111 & 000 \\
111 & 000 & 111 & 000 & 000 & 111 & 000 & 111 & 000 & 000 & 000 & 000 \\
111 & 000 & 111 & 000 & 111 & 000 & 111 & 000 & 000 & 000 & 000 & 000 \\
000 & 111 & 111 & 111 & 111 & 111 & 000 & 000 & 000 & 111 & 000 & 000 \\
111 & 111 & 000 & 111 & 111 & 111 & 000 & 000 & 111 & 000 & 000 & 000
\end{array}\right]$$

in which the triples of coordinates in every codeword are indistinguishable. By projecting the code on two coordinates (puncturing every third column), we obtain code $\mathcal{F}$.

$$G_{\mathcal{F}} = \begin{bmatrix} 11 & 00 & 11 & 11 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 11 \\ 11 & 11 & 11 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 11 & 00 \\ 11 & 00 & 11 & 00 & 00 & 11 & 00 & 11 & 00 & 00 & 00 & 00 \\ 11 & 00 & 11 & 00 & 11 & 00 & 11 & 00 & 00 & 00 & 00 & 00 \\ 00 & 11 & 11 & 11 & 11 & 11 & 00 & 00 & 00 & 11 & 00 & 00 \\ 11 & 11 & 00 & 11 & 11 & 11 & 00 & 00 & 11 & 00 & 00 & 00 \end{bmatrix}$$

Lemma 4.14. *The dimension of code $\mathcal{F} \subseteq \mathcal{G}$ is 6. Furthermore, this code is unique up to equivalence.*

Proof. By Lemma 4.6, the dimension of the code fixed by ζ in $\mathcal{C}$ is at least 6. We exhaustively verified that for every code $\mathcal{L} \in \mathbb{L}$, the dimension of subcode $\mathcal{F}$ obtained from $\mathcal{L}$ using Lemma 4.12 never exceeds 6. Furthermore, all the obtained codes were equivalent to the code generated by matrix $G_{\mathcal{F}}$ presented in Example 4.13. □

The desired code $\mathcal{G} \simeq \mathcal{G}_{24}$ has to be self-orthogonal, i.e. $\mathcal{F} \subseteq \mathcal{G} \subseteq \mathcal{F}^{\perp}$. Therefore, the remaining 6 vectors must belong to the quotient space $\mathcal{F}^{\perp}/\mathcal{F}$. The code $\mathcal{F}^{\perp}$ has dimension 24 - 6 = 18, and the quotient space $\mathcal{F}^{\perp}/\mathcal{F}$ has dimension 12.

4.2.3 Required Form of Generator Matrix

The first step of our search aims to find generator matrices for all the compatible codes $\mathcal{G}$. It would be rather counter-productive to find generator matrices that produce the same code. Therefore to reduce the size of the search, we specify the exact form of generator matrix that we are looking for.

Definition 4.15 (Basic column)**.** Basic column *b refers to a column of generator matrix G for code $\mathcal{R}$ in which a single* 1 *appears. If there are two columns b, c with a single* 1 *which appears in the same row for both b and c, only one of them can be considered basic.*

Let B be a set of basic columns in generator matrix G for an (n, k, d) linear code $\mathcal{R}$. Clearly, G can be always transformed to a form with k basic columns, which we denote by $b_1, b_2, \ldots, b_k$. We call such generator matrix *maximal*.

Example 4.16. *All of the matrices G_2, G_3 and G_4 generate the same code $\mathcal{R}$ equivalent to code $\mathcal{S}$ generated by matrix G_1. The generator matrix G_1 is in the standard form $[I|P]$ while the remaining matrices are not. The generator matrices G_1, G_2 and G_3 are maximal; the generator matrix G_4 is not.*

$$G_1 = \begin{bmatrix} 1000110 \\ 0100101 \\ 0010011 \\ 0001111 \end{bmatrix} \qquad G_2 = \begin{bmatrix} 1100100 \\ 0000111 \\ 0110001 \\ 0101101 \end{bmatrix}$$

$$G_3 = \begin{bmatrix} 1100100 \\ 0101101 \\ 0110001 \\ 0000111 \end{bmatrix} \qquad G_4 = \begin{bmatrix} 1100100 \\ 0000111 \\ 0110110 \\ 0101101 \end{bmatrix}$$

Each subset of B defines a unique vector in code $\mathcal{R}$. In particular, each individual $b \in B$ defines a unique vector, which is a row of G. Hence, Lemma 4.17 immediately follows.

Lemma 4.17. *A vector* $\mathbf{v}$ *is a row in a maximal generator matrix G if and only if it has exactly one non-zero entry in basic columns.*

It is generally known that every code $\mathcal{R}$ is equivalent to a code $\mathcal{S}$ with a generator matrix in the standard form. However, the code $\mathcal{R}$ itself may not have such a generator matrix. Hence, we cannot require the found generator matrices to be in the standard form. We can, however, require the generator matrices to be maximal. Furthermore, we can specify which columns are to be basic.

In Section 4.2.2, we describe how subcode $\mathcal{F} \subseteq \mathcal{G}$ can be computed. The dimension of $\mathcal{F}$ is 6, thus its maximal generator matrix contains 6 basic columns. The pairs of coordinates in this code are indistinguishable. Therefore, every basic column is "duplicated" (see generator matrix F for code $\mathcal{F}_1^1$ below and columns 13 through 24 as an example).

$$F = \begin{bmatrix} 110011000011001100000000 \\ 110011001100110000000000 \\ 110011110000000000000011 \\ 111111000000000000001100 \\ 001111111111000000110000 \\ 111100111111000011000000 \end{bmatrix} \quad F' = \begin{bmatrix} 000000110000000000000000 \\ 000011000000000000000000 \\ 001100000000000000000000 \\ 110000000000000000000000 \\ 000000000000001100000000 \\ 000000000000110000000000 \\ 010001000001000100000000 \\ 010001000100010000000000 \\ 010001010000000000000001 \\ 010101000000000000000100 \\ 000101010000010100010000 \\ 010100010000010101000000 \end{bmatrix}$$

Consider the 6 rows that are to be filled in the generator matrix of code $\mathcal{F}$ to complete code $\mathcal{G} \simeq \mathcal{G}_{24}$. These rows generate a code $co\mathcal{F}$ such that $\mathcal{F} \oplus co\mathcal{F} = \mathcal{G}$, and they all have to belong to the quotient space $\mathcal{F}^{\perp}/\mathcal{F}$. Matrix F' generates the quotient space corresponding to code $\mathcal{F}_1^1$. In this particular example, there are 6 all-zero columns in F' (columns 9,11,17,19,21,23 read in left-to-right direction) always followed by a basic column. This structure, together with Lemma 4.18, leads to Lemma 4.19.

Lemma 4.18. *Let $\mathcal{G}$ be a code equivalent to the extended Golay code $\mathcal{G}_{24}$ such that $\pi_{\sigma}^{-1}(\mathcal{G})$ is a subcode of a putative $(72, 36, 16)$ Type II code $\mathcal{C}$ as described so far. Then code $\mathcal{G}$ must be fixed by automorphism $\alpha = (1, 2)(3, 4) \ldots (23, 24)$.*

Proof. Analogous to proof of Proposition 4.7, also [27]. □

Lemma 4.19. *For every code $\mathcal{G}$ in the case of code $\mathcal{F}_1^1$, there is a maximal generator matrix G with the basic columns $\{i, i+1 \mid i \in \{9, 11, 17, 19, 21, 23\}\}$.*

Proof. By Lemma 4.18, code $\mathcal{G}$ is fixed by automorphism α. Thus, if column i is basic, column $i + 1$ is also basic. Consider a codeword $\mathbf{w} \in \mathcal{G}$ which has pairs of coordinates indistinguishable. For every such codeword $\mathbf{w}$ and its pair of indistinguishable coordinates $j, j+1$, there must be a codeword $\mathbf{x} \in \mathcal{G}$ such that $\mathbf{x} + \alpha(\mathbf{x}) = \mathbf{w}$ (otherwise $\mathcal{G}$ would contain at least two identical columns). Obviously, $\mathbf{w} \neq \mathbf{x}$. There is a generator matrix for F'' for code $\mathcal{F}_1^1$ with basic columns $9, 11, 17, 19, 21, 23$ (see below). For each codeword

$\mathbf{w} \in F''$, the codeword $\mathbf{x}$ such that $\mathbf{x} + \alpha(\mathbf{x}) = \mathbf{w}$ belongs to the quotient space generated by F'. As all the codewords in this quotient space have entry 0 at all of the positions $i+1 \in \{10, 12, 18, 20, 22, 24\}$, all the columns $i, i+1$ can be required to be basic.

$$F'' = \begin{bmatrix} 110011000011001100000000 \\ 110011001100110000000000 \\ 110011110000000000000011 \\ 111111000000000000001100 \\ 001111110000111100110000 \\ 111100110000111111000000 \end{bmatrix}$$

□

During our computations, we always have the generator matrix for $\mathcal{F}$ available. In every case, we can see that the structure of the code generated by $\mathcal{F}$ and the corresponding quotient space is similar to the one presented in this section. Therefore, Lemma 4.19 can be adjusted analogously using the same arguments.

4.2.4 The Algorithm

Based on the concepts introduced in the previous sections, we can summarize the algorithm searching for a generator matrix of $(72, 24, 16)$ subcode of $\mathcal{C}$ as follows:

Step 1: For each conjugacy class representative of a self-dual codes $\mathcal{L}_i \in \mathbb{L}$ and its automorphism σ_j^i of order 3 do the following

a) Reorganize code $\mathcal{L}_i$ so that σ_j^i acts as $(1, 2, 3) \dots (34, 35, 36)$ (cf. Section 4.1.1)

b) Find the set $\mathfrak{R}$ of vectors compatible with $\mathcal{L}_i$ (cf. Section 4.2.1)

c) Find the subcode $\mathcal{F} \subseteq \mathcal{G}$ (cf. Section 4.2.3)

d) Find the quotient space $\mathfrak{V} = \mathcal{F}/\mathcal{F}^\perp$ (Section 4.2.2)

e) Find the set $\mathfrak{W}$ of candidates from $\mathfrak{R} \cup \mathfrak{V}$ such that all the vectors in $\mathfrak{W}$ are in the form admitting a maximal generator matrix for $\mathcal{G}$ with specified basic columns as described in Section 4.2.3

f) Exhaustively enumerate the generator matrices for all the codes $\mathcal{G} \simeq \mathcal{G}_{24}$ in the required form that consist of generator matrix for the subcode $\mathcal{F}$ completed by 6 vectors from $\mathfrak{W}$. Use them to create a set $S_{\mathcal{L}_i}$ of generator matrices for the extended codes $\pi_\theta^{-1}(\mathcal{L}_i) + \pi_\sigma^{-1}(\mathcal{G})$.

Step 2: Find the pairwise non-equivalent code representatives from $\bigcup_{\mathcal{L}_i \in \mathbb{L}} S_{\mathcal{L}_i}$.

4.2.5 The Results

Appendix A provides a list of generator matrices for the family of codes $\mathbb{L} = \{\mathcal{L}_i \mid 1 \leq i \leq 41\}$. There are in total 19 conjugacy classes of a code $\mathcal{L}_i$ and automorphism of order 3 without any fixed points representatives of which are listed below. Automorphism σ_r^i denotes an automorphism of code $\mathcal{L}_i$; subscript r is to distinguish among multiple such automorphism of a single code.

$$
\begin{aligned}
\sigma_1^1 &= (1,18,29)(2,14,26)(3,24,28)(4,13,16)(5,33,7)(6,32,12)(8,10,9)\\
&\quad (11,15,31)(17,30,34)(19,36,27)(20,23,25)(21,35,22)\\
\sigma_1^3 &= (1,36,15)(2,20,3)(4,21,23)(5,13,16)(6,12,10)(7,8,9)(11,25,17)\\
&\quad (14,29,31)(18,32,22)(19,30,26)(24,34,33)(27,28,35)\\
\sigma_1^6 &= (1,5,7)(2,15,13)(3,12,21)(4,35,14)(6,33,18)(8,23,10)(9,31,20)\\
&\quad (11,26,27)(16,22,29)(17,24,36)(19,30,25)(28,34,32)\\
\sigma_1^7 &= (1,36,22)(2,7,34)(3,23,19)(4,26,11)(5,28,8)(6,21,16)(9,17,35)\\
&\quad (10,33,25)(12,30,14)(13,27,29)(15,32,18)(20,31,24)\\
\sigma_1^8 &= (1,35,21)(2,10,17)(3,15,16)(4,32,22)(5,20,25)(6,11,29)(7,30,36)\\
&\quad (8,24,33)(9,34,28)(12,14,31)(13,19,18)(23,27,26)\\
\sigma_1^9 &= (1,10,20)(2,6,30)(3,11,5)(4,13,22)(7,15,33)(8,24,32)(9,19,18)\\
&\quad (12,25,23)(14,34,26)(16,35,21)(17,28,29)(27,36,31)\\
\sigma_1^{11} &= (1,19,13)(2,11,15)(3,24,6)(4,9,27)(5,20,25)(7,30,36)(8,29,18)\\
&\quad (10,35,16)(12,28,32)(14,26,31)(17,21,33)(22,23,34)\\
\sigma_1^{12} &= (1,11,22)(2,5,16)(3,34,10)(4,14,33)(6,18,35)(7,13,26)(8,28,30)\\
&\quad (9,24,17)(12,23,19)(15,29,27)(20,31,36)(21,25,32)
\end{aligned}
$$

$$
\begin{aligned}
\sigma_1^{14} &= (1,2,19)(3,21,15)(4,9,12)(5,36,20)(6,24,13)(7,25,30)(8,18,35)\\
&\quad (10,16,29)(11,33,17)(14,31,34)(22,28,23)(26,32,27)\\
\sigma_1^{16} &= (1,23,24)(2,9,28)(3,11,6)(4,30,10)(5,25,31)(7,27,17)(8,19,12)\\
&\quad (13,16,15)(14,35,33)(18,34,32)(20,36,29)(21,22,26)\\
\sigma_1^{21} &= (1,15,22)(2,21,26)(3,7,32)(4,17,6)(5,23,24)(8,27,10)(9,28,11)\\
&\quad (12,13,30)(14,16,35)(18,19,36)(20,25,29)(31,34,33)\\
\sigma_2^{21} &= (1,22,15)(2,26,21)(3,32,7)(4,17,6)(5,24,23)(8,27,10)(9,28,11)\\
&\quad (12,13,30)(14,35,16)(18,19,36)(20,29,25)(31,34,33)\\
\sigma_1^{22} &= (1,29,30)(2,15,26)(3,25,6)(4,17,23)(5,19,18)(7,11,32)(8,12,14)\\
&\quad (9,13,10)(16,21,20)(22,24,35)(27,28,36)(31,33,34)\\
\sigma_2^{22} &= (1,20,24)(2,10,12)(3,7,31)(4,27,19)(5,23,36)(6,32,34)(8,26,13)\\
&\quad (9,14,15)(11,33,25)(16,35,29)(17,28,18)(21,22,30)\\
\sigma_3^{22} &= (1,7,33)(2,27,18)(3,21,24)(4,13,12)(5,15,28)(6,16,22)(8,23,9)\\
&\quad (10,14,17)(11,34,29)(19,26,36)(20,35,25)(30,32,31)\\
\sigma_1^{25} &= (1,4,17)(2,26,12)(3,15,20)(5,11,23)(6,35,14)(7,36,32)(8,22,30)\\
&\quad (9,21,31)(10,29,18)(13,28,19)(16,33,25)(24,27,34)\\
\sigma_2^{25} &= (1,4,17)(2,26,12)(3,15,20)(5,11,23)(6,35,14)(7,32,36)(8,30,22)\\
&\quad (9,31,21)(10,18,29)(13,28,19)(16,25,33)(24,34,27)\\
\sigma_3^{25} &= (1,2,15)(3,17,12)(4,26,20)(5,14,28)(6,19,11)(7,25,18)(8,31,27)\\
&\quad (9,34,22)(10,36,16)(13,23,35)(21,24,30)(29,32,33)\\
\sigma_4^{25} &= (1,12,20)(2,3,4)(5,35,19)(6,28,23)(7,33,10)(8,21,34)(9,27,30)\\
&\quad (11,14,13)(15,17,26)(16,18,32)(22,31,24)(25,29,36)
\end{aligned}
$$

Thus, there are in total 19 cases to process by the algorithm presented in Section 4.2.4 (Step 1). We have carried out the computations.

For each case, Appendix B provides the code $\mathcal{D}$ that extends to code $\mathcal{F}$—subcode of $\mathcal{G}$ by doubling the coordinates (Step 1c). Interestingly, all the codes $\mathcal{F}$ are equivalent to the (24,6,8) code generated by the following matrix:

$$
G = \begin{bmatrix}
110011110000000000000011\\
111111000000000000001100\\
110011000011001100000000\\
110011001100110000000000\\
001111111111000000110000\\
111100111111000011000000
\end{bmatrix}
$$

In total, we found 36 pairwise non-equivalent codes $\mathcal{X} = \pi_\theta^{-1}(\mathcal{L}) + \pi_\sigma^{-1}(\mathcal{G})$, generator matrices of which are listed in Appendix C. We denote this codes by $\mathcal{X}_i$ with $1 \leq i \leq 36$. Table 4.2 provides a summary of the cases from which the codes were created.

Case	Code and automorphism (cf. Step 1a)	Created codes
1	$\mathcal{L}_1$ reorganized with respect to σ_1^1	$\mathcal{X}_1$, $\mathcal{X}_2$
2	$\mathcal{L}_3$ reorganized with respect to σ_1^3	$\mathcal{X}_3$, $\mathcal{X}_4$
3	$\mathcal{L}_6$ reorganized with respect to σ_1^6	$\mathcal{X}_5$, $\mathcal{X}_6$, $\mathcal{X}_7$
4	$\mathcal{L}_7$ reorganized with respect to σ_1^7	$\mathcal{X}_8$,$\mathcal{X}_9$
5	$\mathcal{L}_8$ reorganized with respect to σ_1^8	$\mathcal{X}_{10}$,$\mathcal{X}_{11}$,$\mathcal{X}_{12}$
6	$\mathcal{L}_9$ reorganized with respect to σ_1^9	$\mathcal{X}_{13}$,$\mathcal{X}_{14}$,$\mathcal{X}_{15}$
7	$\mathcal{L}_{11}$ reorganized with respect to σ_1^{11}	$\mathcal{X}_{16}$,$\mathcal{X}_{17}$,$\mathcal{X}_{18}$
8	$\mathcal{L}_{12}$ reorganized with respect to σ_1^{12}	$\mathcal{X}_{19}$,$\mathcal{X}_{20}$
9	$\mathcal{L}_{14}$ reorganized with respect to σ_1^{14}	$\mathcal{X}_{21}$,$\mathcal{X}_{22}$
10	$\mathcal{L}_{16}$ reorganized with respect to σ_1^{16}	$\mathcal{X}_{23}$,$\mathcal{X}_{24}$
11	$\mathcal{L}_{21}$ reorganized with respect to σ_1^{21}	$\mathcal{X}_{25}$
12	$\mathcal{L}_{21}$ reorganized with respect to σ_2^{21}	$\mathcal{X}_A \simeq \mathcal{X}_{25}$
13	$\mathcal{L}_{22}$ reorganized with respect to σ_1^{22}	no compatible code $\mathcal{G}$ exists
14	$\mathcal{L}_{22}$ reorganized with respect to σ_2^{22}	$\mathcal{X}_{26}$, $\mathcal{X}_{27}$, $\mathcal{X}_{28}$
15	$\mathcal{L}_{22}$ reorganized with respect to σ_3^{22}	$\mathcal{X}_{29}$, $\mathcal{X}_{30}$, $\mathcal{X}_{31}$
16	$\mathcal{L}_{25}$ reorganized with respect to σ_1^{25}	$\mathcal{X}_{32}$, $\mathcal{X}_{33}$
17	$\mathcal{L}_{25}$ reorganized with respect to σ_2^{25}	$\mathcal{X}_{34}$, $\mathcal{X}_{35}$
18	$\mathcal{L}_{25}$ reorganized with respect to σ_3^{25}	$\mathcal{X}_{36}$
19	$\mathcal{L}_{25}$ reorganized with respect to σ_4^{25}	$\mathcal{X}_B \simeq \mathcal{X}_{36}$

Table 4.2: Scheme of the generating process.

4.3 Step 2: Completing the Generator Matrix

Before describing the next step of the search, let us restate the properties of the codes $\mathcal{X} \in \bigcup_{i=1}^{36} \mathcal{X}_i$:

Proposition 4.20. *Let $\mathcal{X} = \pi_\theta^{-1}(\mathcal{L}) + \pi_\sigma^{-1}(\mathcal{G})$ be a subcode of a putative Type II code $\mathcal{C}$ obtained from a permutation $\mathcal{G}$ of the extended Golay code and a $(36, 18, 8)$ self-dual code $\mathcal{L}$ as described so far. The code $\mathcal{X}$ has the following properties:*

(a) $\mathcal{X}$ is a $(72, 24, 16)$ code

(b) $\mathcal{X}$ is fixed by automorphism ζ

(c) $\mathcal{X}$ is self-orthogonal and is not self-dual

(d) $\mathcal{X} \subseteq \mathcal{C} \subseteq \mathcal{X}^\perp$

As presented in Section 4.2.5, the properties of Proposition 4.20 are not contradictory and such codes $\mathcal{X}$ exist. Thus, we need to fill in up to 12 additional vectors generating component $\mathcal{Q}$ of code $\mathcal{C}$ into generator matrix of $\mathcal{X}$ in order to either find the generator matrix for code $\mathcal{C}$, or to show that $\mathcal{C}$ does not exist.

First, we are going to show that in addition to properties stated by Proposition 4.20, the codes $\mathcal{X} \in \bigcup_{i=1}^{36} \mathcal{X}_i$ obtained so far have a very regular structure.

4.3.1 Structural Analysis

Consider generator matrix G_1 for code $\mathcal{X}_1$ below as an example. The span of this matrix, code $\mathcal{X}_1$, is fixed by automorphism ζ. This automorphism permutes 6-tuples of columns. We denote these 6-tuples by $t_1, t_2, \ldots, t_{12}$. As the permuted columns always circulate within only one 6-tuple t_i, it is only natural to focus on every t_i separately (as indicated by the spacing in G_1). It is easy to verify that every 6-tuple of columns t_i can be row-reduced to a $(6, 4, 2)$ code generated by matrix M.

$G_1 =$

t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}
100011	000000	000000	000000	000000	000000	101010	010101	010101	001001	100011	110001
010010	000000	000000	000000	000000	000000	111111	111111	111111	101101	010010	001001
001001	000000	000000	000000	000000	000000	111111	111111	111111	110110	001001	100100
000111	000000	000000	000000	000000	000000	010101	101010	101010	010010	000111	100011
000000	100011	000000	000000	000000	000000	101101	010010	011100	100011	010101	100100
000000	010010	000000	000000	000000	000000	011011	011011	010010	010010	111111	110110
000000	001001	000000	000000	000000	000000	101101	101101	001001	001001	111111	011011
000000	000111	000000	000000	000000	000000	011011	100100	111000	000111	101010	001001
000000	000000	100011	000000	000000	000000	001110	001110	010010	101010	111000	100011
000000	000000	010010	000000	000000	000000	001001	001001	011011	111111	100100	010010
000000	000000	001001	000000	000000	000000	100100	100100	101101	111111	010010	001001
000000	000000	000111	000000	000000	000000	011100	011100	100100	010101	110001	000111
000000	000000	000000	100011	000000	000000	001001	111111	010101	001110	011011	010101
000000	000000	000000	010010	000000	000000	101101	000000	111111	001001	110110	111111
000000	000000	000000	001001	000000	000000	110110	000000	111111	100100	011011	111111
000000	000000	000000	000111	000000	000000	010010	111111	101010	011100	110110	101010
000000	000000	000000	000000	100011	000000	010101	000000	100011	100011	001001	111111
000000	000000	000000	000000	010010	000000	111111	000000	010010	010010	101101	000000
000000	000000	000000	000000	001001	000000	111111	000000	001001	001001	110110	000000
000000	000000	000000	000000	000111	000000	101010	000000	000111	000111	010010	111111
000000	000000	000000	000000	000000	100011	111111	100011	101010	001110	100100	000000
000000	000000	000000	000000	000000	010010	000000	010010	111111	001001	110110	000000
000000	000000	000000	000000	000000	001001	000000	001001	111111	100100	011011	000000
000000	000000	000000	000000	000000	000111	111111	000111	010101	011100	001001	000000

$$M = \begin{bmatrix} 100011 \\ 010010 \\ 001001 \\ 000111 \end{bmatrix}$$

The block M repeatedly occurs in the generator matrix G_1. The $(6, 4, 2)$ code $\mathcal{M}$ generated by M admits itself automorphism $\zeta' = (1, 2, 3, 4, 5, 6)$. This automorphism must be preserved even after extending the generator matrix G_1, and consequently M, with the remaining rows.

The space $\mathcal{M}$ generated by matrix M is preserved by ζ'. Suppose that we introduce a single additional row to M, creating a matrix N. Either N also generates $\mathcal{M}$, or the space generated by N is not preserved by ζ'. The only option to preserve ζ' is not to add anything, or to add two linearly independent codewords which increase the dimension of $\mathcal{M}$ to 6. Thus, we can guarantee the following:

Lemma 4.21. *Let t_i be a consecutive 6-tuple of columns that ζ permutes among themselves. Then, either all the codewords in $\mathcal{C}$ have in t_i pattern that belongs to $\mathcal{M}$, or t_i is formed by 6 distinct codewords whose patterns make a 6×6 identity block.*

Based on the previous notion, the vectors of $\mathbb{GF}(2)^6$ can be divided into the following two groups:

Group 1: *Non-extending vectors* are the vectors in $\mathcal{M}$

Group 2: *Extending vectors* are the vectors in $\mathbb{GF}(2)^6 \backslash \mathcal{M}$

Similarly, we call the vectors of $\mathbb{GF}(2)^{72}$ *extending* if they contain at least one extending segment of length 6, and *non-extending* otherwise.

Without loss of generality, we can pose the requirement of maximality (see Section 4.2.3) on the desired generator matrix of code $\mathcal{C}$. Let us do a case analysis of how the 36 basic columns can be distributed over the column 6-tuples. A column 6-tuple where all the columns are basic will be called *complete* and $G^\times$ will denote a generator matrix for some code $\mathcal{X}_i$ for $1 \leq i \leq 36$ obtained in the previous phase of the search.

Case 1 (No Identity Block)**.** *For every maximal generator matrix G of the desired code $\mathcal{C}$, no 6-tuple t is complete.*

Lemma 4.22. *Let G be a maximal generator matrix satisfying Case 1. Then G must be obtained from $G^\times$ by adding non-extending codewords only.*

Proof. Assume for contradiction that an extending codeword is added to the matrix $G^\times$ to obtain G. As argued above, the code generated by G contains 6 codewords that extended some 6-tuple t and that all together form a 6×6 identity block on t. Let us form a generator matrix G' by selecting those 6 codewords, let the form set R, and arbitrary remaining 30 codewords. The codewords from R can be used to eliminate all the 1 entries in the remaining rows in 6-tuple t so that the 6×6 identity block is the only block on t that is not all-zero. This 6-tuple t suddenly contains 6 basic columns, i.e. is complete—a contradiction. □

Case 2 (1 Identity Block)**.** *There is a maximal generator matrix G for code $\mathcal{C}$ in which exactly 6 basic columns belong to the same 6-tuple t_i, and no other 6 basic columns belong to the same 6-tuple t_j.*

In this case, 6-tuple t_i must be extended by two extending vectors $\mathbf{w}$ and $\zeta(\mathbf{w})$. By Lemma 4.21, we can insist on G having an identity block in t_i.

Lemma 4.23. *Let G be a maximal generator matrix that satisfies Case 2. The last 10 rows of G must either be formed by non-extending vectors only, or there is a generator matrix G' for $\mathcal{C}$ with more than one 6-tuple that contains 6 basic columns.*

Proof. Consider the added rows of G. We enforce identity on t_i, i.e. we can guarantee that the $\mathbf{w}$ and $\zeta(\mathbf{w})$ have patterns [000010] and [000001] resp. in t_i. Since all columns in t_i are basic, the remaining 10 words have all-zero patterns in t_i. With respect to 6-tuple t_j, there are only 4 options for the configurations of the words $\mathbf{w}$, $\zeta(\mathbf{w})$ and $\mathbf{x}$—one of the remaining 10 words. We denote extending and non-extending sections by E and N respectively.

(a) Besides 6-tuple t_i, no 6-tuple t_j is extended by $\mathbf{x}$:

		t_i		t_j	
$\mathbf{w}:$	...	$\lvert 000010 \rvert$	...	$\lvert N \rvert$	...
$\zeta(\mathbf{w}):$	...	$\lvert 000001 \rvert$	...	$\lvert N \rvert$	...
$\mathbf{x}:$	...	$\lvert 000000 \rvert$	...	$\lvert N \rvert$	...

or

		t_i		t_j	
$\mathbf{w}:$	...	$\lvert 000010 \rvert$	...	$\lvert E \rvert$	...
$\zeta(\mathbf{w}):$	...	$\lvert 000001 \rvert$	...	$\lvert E \rvert$	...
$\mathbf{x}:$	...	$\lvert 000000 \rvert$	...	$\lvert N \rvert$	...

These cases do not violate Lemma 4.23.

(b) Codeword $\mathbf{x}$ is an extending word:

		t_i		t_j	
$\mathbf{w}:$	...	$\lvert 000010 \rvert$	...	$\lvert N \rvert$	...
$\zeta(\mathbf{w}):$	...	$\lvert 000001 \rvert$	...	$\lvert N \rvert$	...
$\mathbf{x}:$	...	$\lvert 000000 \rvert$	...	$\lvert E \rvert$	...

or

		t_i		t_j	
$\mathbf{w}:$	...	$\lvert 000010 \rvert$	...	$\lvert E \rvert$	...
$\zeta(\mathbf{w}):$	...	$\lvert 000001 \rvert$	...	$\lvert E \rvert$	...
$\mathbf{x}:$	...	$\lvert 000000 \rvert$	...	$\lvert E/N \rvert$	...

The codewords $\mathbf{w}$ and $\zeta(\mathbf{w})$ are used to "clean" the block t_i in $\mathbf{x}$. Depending on the form of $\mathbf{w}$ and $\zeta(\mathbf{w})$, $\mathbf{x}$ can remain extending (always the former case) or can become non-extending (possible in the latter case). If the added codeword becomes extending, two independently extended blocks t_i and t_j exist. Both of them can be required to contain 6×6 identity in a maximal generator matrix G'.

□

Case 2 and Lemma 4.23 can be easily generalized:

Case 3 (General)**.** *There is a maximal generator matrix G for code $\mathcal{C}$ in which exactly $6k$, $1 \leq k \leq 5$, basic columns belong to k complete 6-tuples $t^1, t^2, \ldots, t^k$, and no other 6 basic columns belong to the same 6-tuple t.*

Lemma 4.24. *Let G be a generator matrix that satisfies Case 3. The last $12 - 2k$ rows must either be formed by non-extending vectors only, or there is a generator matrix G' for $\mathcal{C}$ with more than k complete 6-tuples.*

Proof. Exhaustive repetition of the argument presented as the proof of Lemma 4.23. □

Case 4 (6 Identity Blocks)**.** *There is a maximal generator matrix G for code $\mathcal{C}$ with 6 complete 6-tuples $t^1, t^2, \ldots, t^6$.*

The algorithm to finish the search must respect the aforementioned cases. It is described in the following section together with the obtained results.

4.3.2 The Algorithm and Results

In the notion of Section 4.3, we now describe the algorithm used to finalize the search for a putative $(72, 36, 16)$ Type II code $\mathcal{C}$. We conduct the search in 7 individual steps for each so far generated code $\mathcal{X} \in \bigcup_{i=1}^{36} \mathcal{X}_i$. In every step, we search for the generator matrices with a fixed number of 6-tuples that contain 6 basic columns. See the case analysis above for further details.

We start with the case when there are no complete column 6-tuples. By Lemma 4.22, we need to select a combination of 12 non-extending candidates that completes code $\mathcal{X}$. For every selected candidate $\mathbf{w}$, we also can count on $\zeta(\mathbf{w})$ being in code $\mathcal{C}$, since $\mathcal{C}$ it is supposed to be fixed by ζ.

Instead of a complete search for 12 codewords, we can first find the set $\mathfrak{C}$ of valid non-extending candidates as follows: For each non-extending codeword $w \in \mathcal{X}/\mathcal{X}^{\perp}$, create a code $\mathcal{C}' = \mathcal{X} \cup \{\mathbf{w}, \zeta(\mathbf{w})\}$. Codeword w belongs to candidates $\mathfrak{C}$ if and only if the minimum distance of $\mathcal{C}'$ is at least 16.

Our computations show that there are no valid non-extending candidates for any of the codes $\mathcal{X} \in \bigcup_{i=1}^{36} \mathcal{X}_i$ as the minimum distance always drops to less than 16. The following is a corollary of this result and Lemmas 4.21 and 4.24:

Corollary 4.25. *Every code $\mathcal{C}$ with automorphism ζ has a generator matrix G composed from some $\mathcal{X} \in \bigcup_{i=1}^{36} \mathcal{X}_i$ and additional 12 extending vectors $\mathbf{w_1}, \mathbf{w_2}, \ldots, \mathbf{w_{12}}$, where $\mathbf{w_{i+6}} = \zeta(\mathbf{w_i})$. Furthermore, there are 6 complete 6-tuples $t^1, t^2, \ldots, t^6$ in G, where t^j is extended by $\mathbf{w_j}$ and $\mathbf{w_{j+6}} = \zeta(\mathbf{w_j})$.*

For a fixed combination of 6-tuples $t^1, t^2, \ldots, t^6$, we can again first find the set of candidates. A valid candidate word $\mathbf{w}$ for a block t^j satisfies the following two conditions:

(1) $\mathbf{w}$ is extending on t^j

(2) for all $i < j$, $\mathbf{w}$ has all-zero pattern on t^i, i.e. w does not extend any 6-tuple on the left from the 6-tuple t^j, and all the required columns are basic

We have verified that for every code $\mathcal{X} \in \bigcup_{i=1}^{36} \mathcal{X}_i$, there is exactly one combination of 6-tuples $t^1, t^2, \ldots, t^6$ which admits non-empty set of valid candidates, denoted by $\mathfrak{C}_i$ for t^i. Thus, the following algorithm finishes the exhaustive search: For each combination of $\mathbf{w_1}, \mathbf{w_2}, \ldots, \mathbf{w_6}$ from $\mathfrak{C}_1, \mathfrak{C}_2, \ldots, \mathfrak{C}_6$ resp. create a sequence of codes $\{\mathcal{C}_i\}_{i=0}^{6}$ where $\mathcal{C}_0 = \mathcal{X}$, $\mathcal{C}_i = \mathcal{C}_{i-1} \cup \{\mathbf{w_i}, \zeta(\mathbf{w_i})\}$ and possibly $\mathcal{C}_6 = \mathcal{C}$. If any of the codes $\mathcal{C}_i$ has minimum distance less than 16, the combination of extending codewords $\mathbf{w_1}, \mathbf{w_2}, \ldots, \mathbf{w_6}$ cannot result in $(72, 36, 16)$ Type II code $\mathcal{C}$.

Given the size of the codes to work with computing the exact minimum distance can easily become excessively time-consuming. Hence for our computations, we used an upper bound on the possible minimum distance $ap(G)$ that is based on a generator matrix G only rather than on an entire code. The upper bound is computed as the minimum weight of a codeword that can be obtained by combination of at most 4 codewords in the generator matrix G. It is easy to see that if $\mathcal{R}$ is an (n, k, d) code with generator matrix G, $d \leq ap(G)$.

Our exhaustive computations show that for all of the generator matrices listed in Appendix C, the minimum weight $d \leq ap(G)$ decreases below 16 when trying to fill in the remaining 12 codewords. All the computations were conducted in a system implemented on our own in C/C++. The overall runtime was under 20 minutes for every case. The equivalence filtering between the first and second phase was done using GAP. The computations were carried out on a 2.4 GHz processor AMD Opteron 6136.

Thus, we can state the following result:

Theorem 4.26. *There is no (72,36,16) Type II code that admits $\mathbb{Z}_6$ as an automorphism group.*

Chapter 5

Becker's Construction of Self-dual Codes

This chapter further investigates the construction of the extended Golay code presented in Section 3.2 of Chapter 3. First, let us formally restate the general scheme of the construction:

Definition 5.1 (Becker's construction)**.** *Let $\mathcal{C}$ be an (n, k, d) linear binary code with a generator matrix G and R set of pairs $\{(\alpha_i, \pi_i) \mid 1 \leq i \leq c\}$, where α_i is a column permutation of G and π_i a projection $\mathbb{GF}(2) \to \mathbb{GF}(2)^c$ mapping 0 to a sequence of 0's. Let us define extended projection $\hat{\pi}_i$ of words $\mathbf{w} \in \mathbb{GF}(2)^n$ as*

$$\hat{\pi}_i(w_1, w_2, \ldots, w_n) = (\pi_i(w_1), \pi_i(w_2), \ldots, \pi_i(w_n)).$$

Then, the code $\mathcal{D}$ resulting from $\mathcal{C}$ by Becker's stacked matrix construction denoted by $(\mathcal{C}, R)$ is generated by a matrix that consists of codewords in set

$$G' = \bigcup_{i=1}^{c} \{\hat{\pi}_i(\alpha_i(\mathbf{w})) \mid \mathbf{w} \in G\}.$$

If $(\mathcal{C}, R)$ is a stacked matrix construction with $\mathcal{C}$ being an (n, k, d) code with generator matrix A and c the cardinality of R, the code $\mathcal{D}$ produced by $(\mathcal{C}, R)$ is a (cn, k', d') code with $k \leq k' \leq ck$ and $1 \leq d' \leq cd$. The generator matrix G' of $\mathcal{D}$ can be written as

$$G = \begin{bmatrix} \hat{\pi}_1(\alpha_1(A)) \\ \hat{\pi}_2(\alpha_2(A)) \\ \cdots \\ \hat{\pi}_c(\alpha_c(A)) \end{bmatrix}$$

It is easy to see that if $\mathcal{C}$ and $\mathcal{D}$ are two equivalent linear binary codes, the constructions $(\mathcal{C}, R)$ and $(\mathcal{D}, R)$ result in equivalent codes as well.

Theorem 5.2. *For every stacked matrix construction $(\mathcal{C}, R)$ producing code $\mathcal{D}$, there is a construction $(\mathcal{C}, R')$ with $\alpha_1 = id$ producing code $\mathcal{D}'$ where codes $\mathcal{D}$ and $\mathcal{D}'$ are equivalent.*

Proof. Consider α_1 of $(\mathcal{C}, R)$ and its inverse α_1^{-1}. The codes $\mathcal{C}$ and $\alpha_1^{-1}(\mathcal{C})$ are equivalent as thus, $(\mathcal{C}, R') = (\alpha_1^{-1}(\mathcal{C}), R)$. □

With respect to the formalized notion, the construction of $\mathcal{G}_{24}$ presented in Chapter 3 is $(\mathcal{H}_8, R)$ where $|R| = 3$, $\mathcal{H}_8$ is the $(8, 4, 4)$ extended Hamming code, $\alpha_1 = id$, $\alpha_2 = \alpha_3 = (8, 7, 6, 5)$ and projections π_1, π_2, π_3:

$$\begin{aligned} \pi_1(0) = \pi_2(0) = \pi_3(0) &= (0,0,0) \\ \pi_1(1) &= (1,1,1) \\ \pi_2(1) &= (1,1,0) \\ \pi_3(1) &= (0,1,1) \end{aligned}$$

5.1 Length 24

Let us focus on other variations of Becker's stacked matrix construction that can possibly result in the extended Golay code $\mathcal{G}_{24}$.

We have exhaustively verified that for the projections π_1, π_2, π_3, the construction results in the code $\mathcal{G}_{24}$ if and only if $\alpha_2 = \alpha_3$. Furthermore, there are 7 cosets of $Aut(\mathcal{H}_8)$ whose representatives taken as permutations $\alpha_2 = \alpha_3$ produce $\mathcal{G}_{24}$ from $\mathcal{H}_8$ with the stacked matrix construction—they are precisely the cosets whose representatives permute 4 coordinates in one cycle, while the remaining coordinates form 4 fixed points.

One arising question is if code $\mathcal{G}_{24}$ can be obtained by Becker's construction from other self-dual or self-orthogonal codes as well. The only option is the self-dual code of length 12 denoted by $\mathcal{B}_{12}$ (see the classification of self-orthogonal codes in [35]), or the other ten $(12, 4, 4)$ self-orthogonal codes of length 12 (see [8]).

Theorem 5.3. *The extended Golay code $\mathcal{G}_{24}$ cannot be constructed from the self-dual code $\mathcal{B}_{12}$ with generator matrix G_{12}.*

Proof. The code $\mathcal{B}_{12}$ contains words of weight 6. Without loss of generality, we may assume that $\alpha_1 = id$. Independently of α_2, both the projections π_1 and π_2 must have $\pi_1(1) = \pi_2(1) = 11$ as the resulting code must be doubly even. Thus, to obtain the extended Golay code, the code $\mathcal{B}_{12} \oplus \alpha_2(\mathcal{B}_{12})$ would have to be a $(12, 12, 4)$ binary linear code. Such a code would contain $2^{12} = |\mathbb{GF}(2)^{12}|$ codewords. Therefore, such a code cannot exist, i.e. the construction cannot result in the extended Golay code. □

$$G_{12} = \begin{bmatrix} 111100000000 \\ 001111000000 \\ 000011110000 \\ 000000111100 \\ 000000001111 \\ 010101010101 \end{bmatrix}$$

Theorem 5.4. *The extended Golay code $\mathcal{G}_{24}$ cannot be constructed from any of the ten $(12, 4, 4)$ self-orthogonal codes.*

Proof. All the codes have a codeword of weight 4. To obtain minimum weight 8 in the constructed code, both π_1 and π_2 must have $\pi_1(1) = \pi_2(1) = 11$. Similarly to above, there would have to exist a $(12, 12, 4)$ code, which is not the case. □

Let us make an observation: certain permutations of $\mathcal{G}_{24}$ cannot be obtained by Becker's construction. An example of such a permutation is $\rho = (1, 4)$ applied to generator matrix N in Section 3.2.

5.2 Length 16

In this section, we present another example of a Type II code that can be obtained via Becker's construction—a code $\mathcal{F}'$ equivalent to $(16, 8, 4)$ code $\mathcal{F}_{16}$ (cf. [35]).

This $(16, 8, 4)$ code is a direct sum of two copies of the extended Hamming code $\mathcal{H}_8$. Its generator matrix G_{16} is as follows:

$$G_{16} = \left[\begin{array}{cccccccc|cccccccc} 1&0&0&0&0&0&0&0&1&1&1&0&0&0&0&0\\ 0&1&0&0&0&0&0&0&1&1&0&1&0&0&0&0\\ 0&0&1&0&0&0&0&0&1&0&1&1&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&1&1&1&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0&0&0&1&1&1&0\\ 0&0&0&0&0&1&0&0&0&0&0&0&1&1&0&1\\ 0&0&0&0&0&0&1&0&0&0&0&0&1&0&1&1\\ 0&0&0&0&0&0&0&1&0&0&0&0&0&1&1&1 \end{array}\right]$$

The matrix G_{16} can be permuted into G'_{16}:

$$G'_{16} = \left[\begin{array}{cccccccc|cccccccc} 1&0&0&0&0&0&0&0&1&0&1&0&1&0&0&0\\ 0&0&1&0&0&0&0&0&1&0&1&0&0&0&1&0\\ 0&0&0&0&1&0&0&0&1&0&0&0&1&0&1&0\\ 0&0&0&0&0&0&1&0&0&0&1&0&1&0&1&0\\ 0&1&0&0&0&0&0&0&0&1&0&1&0&1&0&0\\ 0&0&0&1&0&0&0&0&0&1&0&1&0&0&0&1\\ 0&0&0&0&0&1&0&0&0&1&0&0&0&1&0&1\\ 0&0&0&0&0&0&0&1&0&0&0&1&0&1&0&1 \end{array}\right]$$

Consider the following shortening projection σ operating on the particular bits of codewords in $\mathcal{F}'$:

$$\begin{array}{rcl} \sigma(01) & \rightarrow & 1 \\ \sigma(10) & \rightarrow & 1 \\ \sigma(00) & \rightarrow & 0 \end{array}$$

The projection must be a homomorphism, so we can also derive the remaining:

$$\sigma(11) \rightarrow 0$$

Applying σ to the codewords of G'_{16}, we obtain the two stacked copies of the generator matrix G for the $(8, 4, 4)$ extended Hamming code $\mathcal{H}_8$.

$$G = \left[\begin{array}{cccc|cccc} 1&0&0&0&1&1&1&0\\ 0&1&0&0&1&1&0&1\\ 0&0&1&0&1&0&1&1\\ 0&0&0&1&0&1&1&1 \end{array}\right]$$

This provides directions for the stacked matrix construction $(\mathcal{H}_8, R)$ of $\mathcal{F}_{16}$ with $R = \{(id, \pi_1), (id, \pi_2)\}$ where

$$\begin{aligned} \pi_1(0) = \pi_2(0) &= (0,0) \\ \pi_1(1) &= (1,0) \\ \pi_2(1) &= (0,1). \end{aligned}$$

5.2.1 Relation to Codes Over Higher Fields

Construction of $\mathcal{F}' \simeq \mathcal{F}_{16}$ by the means of stacking matrices has an interesting connection to the work of Pasquier [30]. Consider the elements of $\mathbb{GF}(4)$ generated by the polynomial $x^2 + x + 1$:

Power	Polynomial	Binary
0	0	00
1	1	01
x	x	10
x^2	$x+1$	11

Treating the generator matrix G'_{16} as a generator matrix for a linear code over $\mathbb{GF}(4)$ with elements represented in binary, it can be rewritten as

$$G^4_{16} = \begin{bmatrix} x & 0 & 0 & 0 & x & x & x & 0 \\ 0 & x & 0 & 0 & x & x & 0 & x \\ 0 & 0 & x & 0 & x & 0 & x & x \\ 0 & 0 & 0 & x & 0 & x & x & x \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

As the rows $\mathbf{r_i}$, $1 \leq i \leq 4$ of generator matrix G^4_{16} can be expressed as $\mathbf{r_i} = x \cdot \mathbf{r_{i+4}}$ (x being a scalar element of $GF(4)$), G^4_{16} can be simply written as

$$G^4_{16} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

and generates the $(8, 4, 4)$ linear code over $\mathbb{GF}(4)$.

The reverse of this construction is a particular case of a general method of creating binary codes from codes over $\mathbb{GF}(2^r)$, which is called the Gray map:

Definition 5.5 (The Gray map)**.** *Let* $B = \{b_1, b_2, \ldots, b_r\}$ *be a basis of* $\mathbb{GF}(2^r)$*. The binary image for* $\phi_B(\mathbf{v})$ *of* $\mathbf{v} = (v_1, v_2, \ldots, v_n)$ *from* $\mathbb{GF}(2^r)^n$ *with respect to a basis* B *is obtained by replacing each component* v_i *of* $\mathbf{v}$ *by* $(v_i^1, v_i^2, \ldots, v_i^r)$ *where* $v_i = \Sigma_{j=1}^{r} v_i^j b_j$*. The map* ϕ_B *is called the* Gray map *with respect to basis* B*.*

It can be shown that for a certain family of bases, the Gray map images of Type II codes over $\mathbb{GF}(2^r)$ always form binary Type II codes [4]. Pasquier in [30] presents constructions for extremal self-dual doubly even binary codes of lengths 8, 16, 24, 32, 40 and 64 via Gray map.

We did not manage to turn any of the Gray map constructions into the construction in the fashion of Becker nor did we find a construction for other interesting Type II codes. However, as the main interest of our research concerned a search for the extremal Type II code of length 72, our effort in this direction was rather minimal. We present both constructions as interesting and, in our opinion, promising concepts. Therefore, we propose Becker's construction and its relation to the Gray maps as a matter of future work, which we discuss further in Chapter 6.

Chapter 6

Conclusions and Future Work

In this chapter, we discuss our results in terms of previous work. We also provide suggestions and some direction for future research.

6.1 Search for the $(72, 36, 16)$ Type II Code

A corollary of Theorem 4.26 is that a putative (72,36,16) Type II code $\mathcal{C}$ cannot admit an automorphism group that contains $\mathbb{Z}_6$ as a subgroup. This, together with the results in [15, 26] and the preceding papers, reduces the possible automorphism groups of code $\mathcal{C}$ to the following:

Theorem 6.1. *If a (72,36,16) Type II code $\mathcal{C}$ exists and admits a non-trivial automorphism group, then $Aut(\mathcal{C})$ must be isomorphic to one of $\mathbb{Z}_2$, $\mathbb{Z}_3$, $\mathbb{Z}_4$, $\mathbb{Z}_5$, $\mathbb{Z}_2 \times \mathbb{Z}_2$, A_4, $\mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_2$, S_3, S_4, or the dihedral group with 8 elements D_4.*

We would like to point out the reference [5] which was brought to our attention by G. Nebe [27] after our presentation at CGTC43[1]. This paper, developed independently and using very different methodology, also shows that there is no $(72, 36, 16)$ Type II code that admits automorphism group $\mathbb{Z}_6$.

The author of [5] uses a some new results of modular representation theory to study the structure of the code. Obtaining similar results, his

[1]The reference is dated March 15, 2012, and was sent to us one day later. Our presentation at CGTC43 in Boca Raton, FL, where we discussed the methodology used in this work, and presented some of the results, was on March 6, 2012.

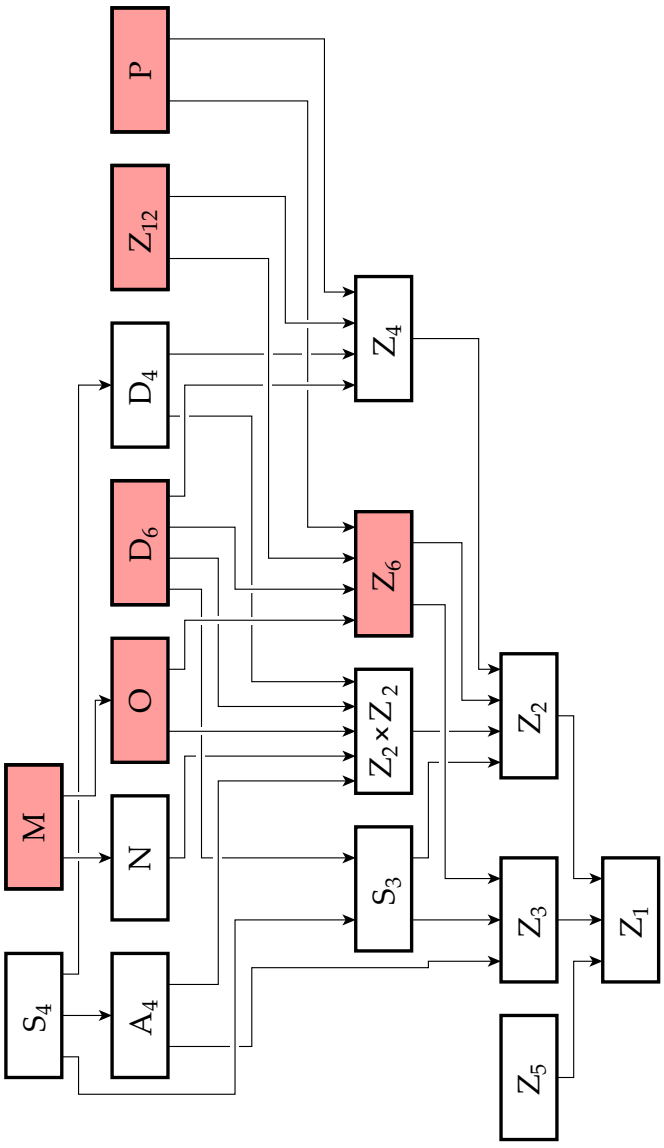

Figure 6.1: Diagram of the permutation groups that can form an automorphism group of a $(72, 36, 16)$ Type II code. The newly eliminated groups are shaded red. The symbol M denotes the group $\mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_3$, N denotes $\mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_2$, O denotes $\mathbb{Z}_2 \times \mathbb{Z}_6$, and P denotes $\mathbb{Z}_3 \rtimes \mathbb{Z}_4$. D_4 denotes the dihedral group on four vertices, which is sometimes also referred to as D_8.

completely algebraic approach also results in some exhaustive computations. These computations are carried out using a general framework for algebraic computations MAGMA [6] unlike in our case, where we developed our own software.

Despite the different methodology, it is possible to observe certain correlation. The following lists the facts that can be followed in both our work and [5]:

- There are 13 codes $\mathcal{L} \in \mathbb{L}$ that admit automorphism of order 3 without any fixed points. In total, there are 18 pairs of some code $\mathcal{L}$ and such an automorphism (up to conjugacy).
- All the codes $\mathcal{D}$ (and consequently codes $\mathcal{F}$—cf. Section 4.1.3 and 4.2.2) are pairwise equivalent.
- According to [5], there are 38 pairwise non-equivalent $(72, 24, 16)$ codes that we denote by $\mathcal{X}_i$, $1 \leq i \leq 36$, $\mathcal{X}_A$ and $\mathcal{X}_B$. We agree that there are 38 such codes when the equivalence is tested individually within the context of each case of code $\mathcal{L} \in \mathbb{L}$ and its fixed-point-free automorphism of order 3. However, there are two additional cases of equivalence across the entire set that [5] missed (see Table 4.2). Therefore, we work with 36 codes only. This however does not affect validity of either of the results.
- There are no $(72, 24, 16)$ codes obtained from the code $\mathcal{L}_{22}$ fixed by automorphism σ_1^{22}.
- None of the codes $\mathcal{X} \in \bigcup_{i=1}^{36} \mathcal{X}_i$ can be completed into the $(72, 36, 16)$ Type II code.

In Chapter 4, Section 4.2.1, we introduced in our opinion a very interesting concept of coordinating two subcodes of code $\mathcal{C}$—that fixing one of them poses strong (weight-based) restrictions on the candidate words for the other code, and how this fact can be effectively used for their enumeration. In our case, we avoided traversing the search space of 24! candidates in the first phase of the search, which we reduced to 2^{12} instead. This significant improvement allowed us to enumerate the set of candidate words and consequently the codes composed from the two initial fragments.

This approach is perhaps applicable to some other cases of the automorphism group and surely is worth further investigation. Recall the special starting position that allowed to use the presented approach: the generator of the assumed automorphism group $\mathbb{Z}_6$ can be decomposed into two distinct permutations, each of which fixes a subcode in the desired code. It does not seem in any way necessary that two distinct permutations have to be provided by decomposing one of the generating permutations. If the assumed automorphism group itself has multiple generators, they introduce the same situation—multiple fixed subcodes.

6.2 Becker's Construction

In Chapter 3, Section 3.2, we presented a new construction of the extended binary Golay code. This notion was formalized and further discussed in Chapter 5.

Without a doubt, Becker's construction is an interesting method of constructing Type II codes, which can provide many answers about their structure. We made a moderate effort to use Becker's construction to create the extended quadratic residue code from Type II codes of length 16. As the search space is enormous and our work was mainly geared towards different results, we did not conduct a full exhaustive search. Our computations produced many $(48, 24, 8)$ codes as well as codes of lower minimum weights, however we did not manage to construct the desired $(48, 24, 12)$ code. Thus, we would like to raise the following question:

Question 6.2. *Can the extended quadratic residue code be obtained by Becker's construction? In particular, can it be obtained from one of the Type II codes of length 16 or the* $(8, 4, 4)$ *extended Hamming code?*

We have also presented a possible connection of Becker's construction to a work of Pasquier and to obtaining binary Type II codes from codes over finite fields with more elements. It would be interesting to show the difference between the two constructions. Betsumiya in [4] showed the construction of binary Type II codes via Gray map cannot produce the extended quadratic residue code from a Type II code of length 8 over $\mathbb{GF}(64)$ for any of the trace-orthogonal bases. Thus, creating $\mathcal{QR}_{48}$ from the extended Hamming code could show a significant difference between the two constructions.

Question 6.3. *Is Becker's construction of Type II codes any different from the construction of binary codes via Gray map?*

The extended quadratic residue code was shown to be unique [18]. The proof, which requires long computations, is the only known argument for the uniqueness. Perhaps when the limits of this construction method are reasonably understood, it can provide us with a simple explanation.

Another problem, which Becker's construction could bring more insight into, is the automorphism group of the extended Golay code. This group is extremely large and we are not aware of any simple explanation of this structure. Any new construction of the Golay code from codes whose automorphism group is much simpler can be of great help in this respect.

Bibliography

[1] E.F. Assmus, Jr., H.F. Mattson, *New 5-designs*, J. Comb. Theory 6 (1969), 122–151.

[2] E.F. Assmus, Jr., H.F. Mattson, Jr., R.J. Turyn, *Research to develop the algebraic theory of codes*, Report AFCRL-67-0365, AirForce Cambridge Res. Labs., Bedford, MA, 1967.

[3] P. Becker, Personal comunication, September 2011 – July 2012.

[4] K. Betsumiya, *Minimum Lee weights of Type II codes over* $\mathbb{F}_{2^r}$, preprint (2001).

[5] M. Borello, *The Automorphism Group of an Extremal [72,36,16] Code does not contain elements of order 6*, arXiv:1203.3321v1 (March 15, 2012).

[6] W. Bosma, J. Cannon, C. Playoust, *The Magma algebra system. I. The user language*, J. Symbolic Comput., 24 (1997), 235–265.

[7] I. Bouyukliev, *About the code equivalence*, Advances in Coding Theory and Cryptology, T. Shaska, W. C. Huffman, D. Joyner, V. Ustimenko, Series (2007) 126–151.

[8] I. Bouyukliev, S. Bouyuklieva, T.A. Gulliver, P.R.J. Östergård *Classification of optimal binary self-orthogonal codes up to length 24*, Journal of Combinatorial Mathematics and Combinatorial Computing 59 (2006), 33–87.

[9] S. Bouyuklieva, *A method for constructing self-dual codes with an automorphism of order 2*, IEEE Trans. Inform. Theory, vol. 46, no. 2 (2000) 496–504.

[10] S. Bouyuklieva, *On the automorphisms of order 2 with fixed points for the extremal self-dual codes of length 24m*, Designs, Codes, Cryptography 25 (2002) 5–13.

[11] S. Bouyuklieva, *On the automorphism group of a doubly-even (72,36,16) code*, IEEE Trans. Inform. Theory 50 (2004), no. 3, 544– 547.

[12] J.H. Conway, N.J.A. Sloane, *A new upper bound on the minimal distance of self-dual codes*, IEEE Trans. Inform. Theory 36 (1990) 13191333.

[13] J.H. Conway, V. Pless, *On primes dividing the group order of a doubly-even (72,36,16) code and the group order of a quaternary (24,12,10) code*, Discrete Math. 38 (1982) 143–156.

[14] S.T. Dougherty, *The Search for the* $(24k, 12k, 4k + 4)$ *Extremal Type II Code*, unpublished manuscript (2011).
Available online: `<http://dl.dropbox.com/u/20879623/survey.pdf>`

[15] T. Feulner, G. Nebe, *The automorphism group of an extremal [72,36,16] code does not contain* $Z_7, Z_3 \times Z_3$, *or* D_{10}, IEEE Transactions of Information Theory, to appear.

[16] J. Fields et al., *Package Guava, GAP – Groups, Algorithms, and Programming, Version 4.5.4*, The GAP Group, 2012.
Available online: `(http://www.gap-system.org)`

[17] M.J.E Golay, *Notes on Digital Coding*, Proc. IRE, vol. 37, (1949) p. 657.

[18] S.K. Houghten, C.W.H. Lam, L.H. Thiel, J.A. Parker, *The extended quadratic residue code is the only (48,24,12) self-dual doubly-even code* IEEE Transactions on Information Theory, vol. 49, no. 1, (2003), 53–59.

[19] S.K. Houghten, M. Letourneau *Optimal Ternary (10,7) Error-Correcting codes*, Congressus Numerantium 155 (2002) 71–80.

[20] W.C. Huffman, V. Pless, *Fundamentals of Error-Correcting Codes*, Cambridge University Press (2003) ISBN 0521782805.

[21] W.C. Huffman, V.Y Yorgov, *A [72,36,16] coubly even code does not have an automorphism of order 11*, IEEE Transactions on Information Theory, vol. 33, no. 5, (1987), 749–752.

[22] F.J. MacWilliams, N.J.A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland (1977), ISBN 0-4448-5193-3.

[23] C.L. Mallows, N.J.A. Sloane, *An upper bound for self-dual codes*, Information and Control 22 (1973), 188–200.

[24] B. McKay, *Nauty*, 2010.
Available online: `<http://cs.anu.edu.au/~bdm/nauty/>`.

[25] C.A. Melchor, P. Gaborit, *On the Classification of Extremal Binary [36,18,8] Self-Dual Codes*, IEEE Transactions on Information Theory, vol. 54, no. 10, (2008).

[26] G. Nebe, *An extremal [72,36,16] binary code has no automorphism group containing $Z_2 \times Z_4$, Q_8, or Z_{10}*, Finite Fields and Their Applications 18 (2012), 563–566.

[27] G. Nebe, *Personal communication*, February – March, 2012.

[28] E.A. O'Brien, W. Willems, *On the automorphism group of a binary self-dual doubly-even [72,36,16] code*, IEEE Trans. Inform. Theory 57 (2011) 4445–4451.

[29] H. Oral, K.T. Phelps, *Almost All Self-Dual Codes Are Rigid*, Journal of Comb. Theory, Ser. A 60 (1992) 264–276.

[30] G. Pasquier, *Binary images of some self-dual codes over $GF(2^m)$ with respect to trace-orthogonal basis*, Discrete Mathematics 37 (1981) 127–129.

[31] G. Pasquier, *A Binary Extremal Doubly Even Self-Dual Code (64,32,12) Obtained from an Extended Reed-Solomon Code over F_{16}*, IEEE Trans. Inform. Theory Vol. 27, No. 6 (1981) 807–808.

[32] V. Pless, *23 Does Not Divide the Order of the Group of a (72,36,16) Doubly-Even Code*, IEEE Trans. Inform. Theory 28 (1982), 113-117.

[33] V. Pless, *Introduction to the Theory of Error-Correcting Codes, 3rd Edition*, John Wiley & Sons (1988), ISBN 0-471-19047-9.

[34] V. Pless, *On the uniqueness of the Golay codes*, J. Comb. Theory 5 (1968), 215–228.

[35] V. Pless, *A classification of self-orthogonal code over GF(2)*, Discrete Mathematics 3, (1972), 209–246.

[36] V. Pless, J.G. Thompson, *17 does not divide the order of the group of a (72,36,16) doubly even code*, IEEE Trans. Inform. Theory Vol. 28, No. 3 (1982) 537–541.

[37] E. Rains, *Shadow Bounds of Self-Dual Codes*.

[38] J. Rosický, *Algebra, 4th edition*, Masaryk University, Brno (2005), ISBN: 80-120-2964-1.

[39] C.E. Shannon, W. Weaver, *The mathematical theory of communication*, Urbana, Illinois: University of Illinois Press (1949).

[40] N.J.A. Sloane, *Is there a* $(72, 36)d = 16$ *self-dual code?*, IEEE Trans. Inform. Theory vol. IT-19 (1973), p. 251 .

[41] N. Yankov, *On the doubly even [72,36,16] codes having an automorphism of order 9*, IEEE Trans. Inform. Theory Vol. 58, No. 1 (2012) 159–163.

Appendix A

List of Binary Self-Dual (36,18,8) Codes

This appendix lists generator matrices for the 41 length-36 codes fixed by σ. The list of generator matrices in [25] contains some discrepancies. The generator matrices used for the search and listed here have been been provided by G. Nebe. We have verified that the code are $[36, 18, 8]$ self-dual doubly even codes. Generator matrix for code $\mathcal{L}_i$ is denoted by G_L^i.

$$G_L^1 = \begin{bmatrix}
100000000000000000011100010101000010 \\
010000000000000000010011100101000010 \\
001000000000000000010110110010000010 \\
000100000000000000000000000111110011 \\
000010000000000000011001111110001101 \\
000001000000000000011101101011111001 \\
000000100000000000000010101001001101 \\
000000010000000000011000100100110011 \\
000000001000000000101100010010010111 \\
000000000100000000101010001011111000 \\
000000000010000001000110111001000000 \\
000000000001000001000110001110111100 \\
000000000000100000101000011100000011 \\
000000000000010000100111101000100000 \\
000000000000001000011101101001011000 \\
000000000000000100000111101000100000 \\
000000000000000010011101010100000001
\end{bmatrix}$$

$$G_L^2 = \begin{bmatrix}
100000000000000000010101011111000110 \\
010000000000000000011010101111000110 \\
001000000000000000000100100111000110 \\
000100000000000000000000000111110011 \\
000010000000000000010000110100100100 \\
000001000000000000010100100010111101 \\
000000100000000000010000111100001001 \\
000000010000000000011111101111111101 \\
000000001000000000110001001001100100 \\
000000000100000000101100100100101111 \\
000000000010000000101010001011111000 \\
000000000001000000100011011001000011 \\
000000000000100000100011000111011100 \\
000000000000010000101000011100000011 \\
000000000000001000010011110100010000 \\
000000000000000100001110110100101100 \\
000000000000000010000011111010000100 \\
000000000000000001011100111111000101
\end{bmatrix}$$

$$G_L^3 = \begin{bmatrix}
100000000000000000111011000000000011\\
010000000000000000110001000001011001 0\\
001000000000000000101001111010110010\\
000100000000000000000000000111110011\\
000010000000000000111011011010011110\\
000001000000000000111010111101111000\\
000000100000000001000111100001111101\\
000000010000000001000111010000111000\\
000000001000000000100011001001100100\\
000000000100000000100111110000100110\\
000000000010000000100011101001001010\\
000000000001000000100011011100100011\\
000000000000100000100011000111011100\\
000000000000010000010100001110000011\\
000000000000001000010011110100010000\\
000000000000000100000101010110011101\\
000000000000000010000001111101000100\\
000000000000000000001011100010110001
\end{bmatrix}$$

$$G_L^4 = \begin{bmatrix}
100000000000000000100110011011111 0010\\
010000000000000000101001101011110010\\
001000000000000000101100001100000011\\
000100000000000000000000000111110011\\
000010000000000000100011100001111101\\
000001000000000000100111001010001001\\
000000100000000001001100000011001111\\
000000010000000001000111010000111000\\
000000001000000000100011111111010101\\
000000000100000000101110100101100110\\
000000000010000000100011110000101 0\\
000000000001000000101100111010010010\\
000000000000100000011001110000101101\\
000000000000010000010100001110000011\\
000000000000001000010011110100010000\\
000000000000000100001110110100100100\\
000000000000000010000011111101000100\\
000000000000000001110101101111100 01
\end{bmatrix}$$

$$G_L^5 = \begin{bmatrix}
100000000000000000101111011100 1001010\\
010000000000000000100000101100 1001010\\
001000000000000000101100001100000011\\
000100000000000000000000000111110011\\
000010000000000000101010111011000101\\
000001000000000000101110100010011 0001\\
000000100000000001001100000011001111\\
000000010000000001000111010000111000\\
000000001000000000011001110000101110\\
000000000100000000010010101110111 10\\
000000000010000000010111001011001010\\
000000000001000000010111111100101 10\\
000000000000100000101000011100000011\\
000000000000010000010011110100010000\\
000000000000001000011101101001011 00\\
000000000000000100000101010110001000\\
000000000000000001010001100101001001
\end{bmatrix}$$

$$G_L^6 = \begin{bmatrix}
100000000000000000111011000000000011\\
010000000000000000110001000001101110\\
001000000000000000101001111011101110\\
000100000000000000000000000111110011\\
000010000000000000111011011010011000\\
000001000000000000111010111101111011\\
000000100000000001000111100000100010\\
000000010000000001000011101000011000\\
000000001000000000110001001001100100\\
000000000100000000100111110001111010\\
000000000010000000100001101000010101\\
000000000001000000100011011100100011\\
000000000000100000100011000111011100\\
000000000000010000010100001110000011\\
000000000000001000010011110100010011\\
000000000000000100000101010111000001\\
000000000000000010000011111101000100\\
000000000000000000001011100011101101
\end{bmatrix}$$

$$G_L^7 = \begin{bmatrix}
100000000000000000101111010010101 0110\\
010000000000000000100000100010101 0110\\
001000000000000000101100001100000011\\
000100000000000000000000000111110011\\
000010000000000000101010111001101 1001\\
000001000000000000101110101100010 1110\\
000000100000000001000111010000111011\\
000000010000000001001101100110010\\
000000001000000000100111010000 1001\\
000000000100000000101111100110101\\
000000000010000000101110100101100\\
000000000001000000101000011100000011\\
000000000000100000100111101000 10011\\
000000000000010000011101101001011 00\\
000000000000001000010011110100010000\\
000000000000000100001110110100100100\\
000000000000000010000011111101000100\\
000000000000000001010001001010101 01
\end{bmatrix}$$

$$G_L^8 = \begin{bmatrix}
100000000000000000101000110101110 0010\\
010000000000000000100111000101110 0010\\
001000000000000000101100001100000011\\
000100000000000000000000000111110011\\
000010000000000000101101011110110 1101\\
000001000000000000101001010110011010\\
000000100000000001001100000011001111\\
000000010000000001000011010000111011\\
000000001000000000101101000100 00110\\
000000000100000000001011110011101 10\\
000000000010000000011010000001 1001\\
000000000001000000000100110111000010\\
000000000000100000000100101100111110\\
000000000000010000010100001110000011\\
000000000000001000010011110100010011\\
000000000000000100001110110100101100\\
000000000000000010000011111101000100\\
000000000000000001001111010111000 01
\end{bmatrix}$$

$$G_L^9 = \begin{bmatrix}
100000000000000000000100111100001 0010\\
010000000000000000000101100110000 1010\\
001000000000000000100111100000010010\\
000100000000000000000000000111110011\\
000010000000000000000101011100111 10\\
000001000000000000000101000010110 1010\\
000000100000000001100111110110111 10\\
000000010000000001110010100101010 10\\
000000001000000000110001001001100100\\
000000000100000000101100010010010111\\
000000000010000000101010001111 1011\\
000000000001000000100011011100100011\\
000000000000100000100011000111011100\\
000000000000010000010100001110000011\\
000000000000001000010011110100010000\\
000000000000000100001110110100101111\\
000000000000000010000011111101000100\\
000000000000000001111111111000010001
\end{bmatrix}$$

$$G_L^{10} = \begin{bmatrix}
100000000000000000001010011100100 0010\\
010000000000000000000101101100100 0010\\
001000000000000000011101010000100 0010\\
000100000000000000000000000111110011\\
000010000000000000000011111001111 0110\\
000001000000000000101110001001110 0111\\
000000100000000001011101110100011 101\\
000000010000000001100010010011001 11\\
000000001000000000101100010010010100\\
000000000100000000101010001011111000\\
000000000010000000100011011100100000\\
000000000001000000100011000111011111\\
000000000000100000101000011100000011\\
000000000000010000010011110100010011\\
000000000000001000011101101001011 11\\
000000000000000100000101111101000100\\
000000000000000001100010111001000001
\end{bmatrix}$$

$$G_L^{11} = \begin{bmatrix}
100000000000000000111001010110011 0110\\
010000000000000000111010011000000 0011\\
001000000000000000101100001100000 0011\\
000100000000000000000000000001111 0011\\
000010000000000000111001111010111 1010\\
000001000000000000111000101000110 1101\\
000000100000000000100100010111111 1001\\
000000010000000000100001101000011 1000\\
000000001000000000110001001001100 0111\\
000000000100000000101000111110100 0010\\
000000000010000000101010001011111 1011\\
000000000001000000101100100000111 0011\\
000000000000100000100011000111011 1111\\
000000000000010000010000100010110 0110\\
000000000000001000010011110100010 0011\\
000000000000000100001010011000011 1010\\
000000000000000010000011111010001 0100\\
000000000000000000000100101100110 1101
\end{bmatrix}$$

$$G_L^{12} = \begin{bmatrix}
100000000000000000100110010101000 0010\\
010000000000000000101001100101000 0010\\
001000000000000000101100001100000 0011\\
000100000000000000000000000001111 0011\\
000010000000000000100011111110000 1101\\
000001000000000000100111010111111 1001\\
000000100000000000100110000001100 1100\\
000000010000000000100001101000011 1011\\
000000001000000000101110011100011 0110\\
000000000100000000101101110000101 0110\\
000000000010000000100001000011110 1010\\
000000000001000000011001101101011 1110\\
000000000000100000101000011100000 0011\\
000000000000010000100111001100011 0011\\
000000000000001000011101101010111 1111\\
000000000000000100000011111010001 0100\\
000000000000000010000001100000111 0011
\end{bmatrix}$$

$$G_L^{13} = \begin{bmatrix}
100000000000000000100101001101011 0110\\
010000000000000000110100011010111 0110\\
001000000000000000111100111110101 1110\\
000100000000000000000000000001111 0011\\
000010000000000000110011100001001 0010\\
000001000000000000100010110110101 1010\\
000000100000000000101000100101100 0010\\
000000010000000000100111111010110 0110\\
000000001000000000110001001001100 0111\\
000000000100000000101100010010010 0100\\
000000000010000000101010001011111 0000\\
000000000001000000100011011100100 0011\\
000000000000100000100011000111011 1111\\
000000000000010000010000111011000 0011\\
000000000000001000001010000111100 0010\\
000000000000000100000011111101000 0111\\
000000000000000010000011101101001 0100\\
000000000000000001000001100110101 1101
\end{bmatrix}$$

$$G_L^{14} = \begin{bmatrix}
100000000000000000100001100010110 0110\\
010000000000000000101111000001011 0110\\
001000000000000000101100001100000 0011\\
000100000000000000000000000001111 0011\\
000010000000000000100001100001100 1001\\
000001000000000000100000001111100 1110\\
000000100000000000100110000001100 1111\\
000000010000000000100001101000011 1011\\
000000001000000000100101011010010 0010\\
000000000100000000011001110000100 0010\\
000000000010000000011111101001001 1101\\
000000000001000000101101111100101 0110\\
000000000000100000101101001011010 1010\\
000000000000010000010111001110000 0011\\
000000000000001000010011110100010 0000\\
000000000000000100001110110100101 1100\\
000000000000000010000111011010000 0111\\
000000000000000001000011010110001 0110\\
000000000000000000100001100110101 1101
\end{bmatrix}$$

$$G_L^{15} = \begin{bmatrix}
100000000000000000101100101111001 1010\\
010000000000000000100011011111001 1010\\
001000000000000000101100001100000 0011\\
000100000000000000000000000001111 0011\\
000010000000000000100100010000110 1010\\
000001000000000000100100010101110 1010\\
000000100000000000101011010011111 1111\\
000000010000000000100101000000000 1111\\
000000001000000000100010100101100 1011\\
000000000100000000100010011010010 1000\\
000000000010000000100001001101001 0110\\
000000000001000000011000010111111 0110\\
000000000000100000000011010010001 1101\\
000000000000010000010100111000001 0101\\
000000000000001000001011100010111 1010\\
000000000000000100001110101101001 0100\\
000000000000000010000111100001011 0011\\
000000000000000001000001110100110 0111
\end{bmatrix}$$

$$G_L^{16} = \begin{bmatrix}
100000000000000000100001001101110 1010\\
010000000000000000101110111011101 1010\\
001000000000000000101100001100000 0011\\
000100000000000000000000000001111 0011\\
000010000000000000100100100110110 1010\\
000001000000000000101001110111001 1010\\
000000100000000000101011010011111 1111\\
000000010000000000100101000000000 1111\\
000000001000000000100010100101100 1011\\
000000000100000000100011010100100 1000\\
000000000010000000101110001101011 0110\\
000000000001000000000011010010001 1101\\
000000000000100000011000010111111 0110\\
000000000000010000100010111010100 0100\\
000000000000001000010000100111001 0010\\
000000000000000100001110101101001 0100\\
000000000000000010000111100001011 0011\\
000000000000000001000001110100110 0111
\end{bmatrix}$$

$$G_L^{17} = \begin{bmatrix}
100000000000000000101000001011111 1110\\
010000000000000000100111111011111 1110\\
001000000000000000101100001100000 0011\\
000100000000000000000000000001111 0011\\
000010000000000000101101100000111 1110\\
000001000000000000100000110001000 1101\\
000000100000000000101011010010111 1100\\
000000010000000000100101000000000 1111\\
000000001000000000100010100101100 1011\\
000000000100000000100010011010010 1000\\
000000000010000000100001100111000 0001\\
000000000001000000010100110111010 1001\\
000000000000100000010100110111010 1010\\
000000000000010000000011010101100 0001\\
000000000000001000000001010101011 1110\\
000000000000000100011101011010100 0111\\
000000000000000010000111100001011 0011\\
000000000000000001001100101111111 1101
\end{bmatrix}$$

$$G_L^{18} = \begin{bmatrix}
100000000000000000101100101101110 1010\\
010000000000000000100011011011101 1010\\
001000000000000000101100001100000 0011\\
000100000000000000000000000001111 0011\\
000010000000000000101001000110110 1001\\
000001000000000000100100100111010 1011\\
000000100000000000101011010010111 1111\\
000000010000000000100101000000000 1111\\
000000001000000000100010100101100 1011\\
000000000100000000100100110101001 1000\\
000000000010000000100001000110101 0110\\
000000000001000000001100001001110 1110\\
000000000000100000000011010111111 1101\\
000000000000010000001010011001110 1110\\
000000000000001000000101110011001 1010\\
000000000000000100011101011010100 0111\\
000000000000000010000111100001011 0011\\
000000000000000001011110111011101 1001
\end{bmatrix}$$

$$G_L^{19} = \begin{bmatrix} 100000000000000000011011101101101010 \\ 010000000000000000000100001101101010 \\ 001000000000000000000000001111000011 \\ 000100000000000000000000001100011011 \\ 000010000000000000011100110011011110 \\ 000001000000000000011101011001001001 \\ 000000100000000000110100001110110001 \\ 000000010000000000101111011111111101 \\ 000000001000000000100010110110110001 \\ 000000000100000000110001001010001100 \\ 000000000010000000101010001100001111 \\ 000000000001000000100101001100010111 \\ 000000000000100000110011001010111111 \\ 000000000000010000101110101000010000 \\ 000000000000001000010101110001100011 \\ 000000000000000100010100000011101100 \\ 000000000000000010001110110001010000 \\ 000000000000000001100111101101101001 \end{bmatrix} \quad G_L^{20} = \begin{bmatrix} 100000000000000000011011110101110010 \\ 010000000000000000000100010101110010 \\ 001000000000000000000000001111000011 \\ 000100000000000000000000001100011011 \\ 000010000000000000011100101011000101 \\ 000001000000000000011101000001010010 \\ 000000100000000000110100010110101001 \\ 000000010000000000101111000111100101 \\ 000000001000000000100010101110101001 \\ 000000000100000000110001001010001100 \\ 000000000010000000101010001100001111 \\ 000000000001000000100101001100010111 \\ 000000000000100000110011001010111111 \\ 000000000000010000101110101000010000 \\ 000000000000001000010101110001100000 \\ 000000000000000100010100000011101111 \\ 000000000000000010001110110001010000 \\ 000000000000000001100111110101110001 \end{bmatrix}$$

$$G_L^{21} = \begin{bmatrix} 100000000000000000111101110110111110 \\ 010000000000000000110001110000000011 \\ 001000000000000000000000001111000011 \\ 000100000000000000111101101111010011 \\ 000010000000000000111101011110101000 \\ 000001000000000000111100111100011100 \\ 000000100000000000110000000100100110 \\ 000000010000000000101001000100101011 \\ 000000001000000000101000010001110010 \\ 000000000100000000100001010111001011 \\ 000000000010000000110010000101011011 \\ 000000000001000000110000100110001100 \\ 000000000000100000100010100111110010 \\ 000000000000010000100111111110100001 \\ 000000000000001000001000100010101110 \\ 000000000000000100001000101101010001 \\ 000000000000000010001011111001001010 \\ 000000000000000001111011011111010000 \end{bmatrix} \quad G_L^{22} = \begin{bmatrix} 100000000000000000101010111011101010 \\ 010000000000000000110101011011101010 \\ 001000000000000000000100100011101010 \\ 000100000000000000111000000110000011 \\ 000010000000000000101001011010011001 \\ 000001000000000000101111011010100110 \\ 000000100000000000011001101101000101 \\ 000000010000000000011000010101111010 \\ 000000001000000000100011011100001011 \\ 000000000100000000100010111010110111 \\ 000000000010000000111010000000011100 \\ 000000000001000000110101000000111011 \\ 000000000000100000101011010100000111 \\ 000000000000010000010010010100011111 \\ 000000000000001000001110110011010100 \\ 000000000000000100001000101101110011 \\ 000000000000000010000000001111011011 \\ 000000000000000001011100111011101001 \end{bmatrix}$$

$$G_L^{23} = \begin{bmatrix} 100000000000000000101010100001000110 \\ 010000000000000000110101000001000110 \\ 001000000000000000000000001111000011 \\ 000100000000000000101101111111111110 \\ 000010000000000000101100011001100101 \\ 000001000000000000000100100100110101 \\ 000000100000000000111100100110111101 \\ 000000010000000000010010010000110101 \\ 000000001000000000110001001100111100 \\ 000000000100000000101000100110101011 \\ 000000000010000000100111101100001100 \\ 000000000001000000011000010010101100 \\ 000000000000100000010111000110010011 \\ 000000000000010000010101111000011100 \\ 000000000000001000010100000011110100 \\ 000000000000000100001101100000000111 \\ 000000000000000010000010101001110100 \\ 000000000000000001010110100001000101 \end{bmatrix} \quad G_L^{24} = \begin{bmatrix} 100000000000000000010100111001111110 \\ 010000000000000000000101101100111110 \\ 001000000000000000011101010000111110 \\ 000100000000000000011111101110011110 \\ 000010000000000000011111110001100110 \\ 000001000000000000011000010111101010 \\ 000000100000000000000010110001110010 \\ 000000010000000000110011111000110110 \\ 000000001000000000101110111001010110 \\ 000000000100000000110110110110001011 \\ 000000000010000000110000101101010011 \\ 000000000001000000100011000010111011 \\ 000000000000100000100111100111111100 \\ 000000000000010000011101101011111100 \\ 000000000000001000001001001111110111 \\ 000000000000000100001000110011001111 \\ 000000000000000010000110011100100111 \\ 000000000000000001100010111001111101 \end{bmatrix}$$

$$G_L^{25} = \begin{bmatrix} 100000000000000000111010010100011110 \\ 010000000000000000110001110000000011 \\ 001000000000000000000000001111000011 \\ 000100000000000000111001111011001110 \\ 000010000000000000111011011000110000 \\ 000001000000000000101001000110011011 \\ 000000100000000000100000110001001101 \\ 000000010000000000100000001011010101 \\ 000000001000000000100010100001110100 \\ 000000000100000000100010010100100110 \\ 000000000010000000110001010001011000 \\ 000000000001000000100010111001101010 \\ 000000000000100000100001100111110001 \\ 000000000000010000010110100110010000 \\ 000000000000001000010100010100111011 \\ 000000000000000100000101000001011101 \\ 000000000000000010000010110100110100 \\ 000000000000000000001000101000111101 \end{bmatrix} \quad G_L^{26} = \begin{bmatrix} 100000000000000000110000100011001010 \\ 010000000000000000110100110000000011 \\ 001000000000000000100111101111001010 \\ 000100000000000000110101001000110100 \\ 000010000000000000110001011100111011 \\ 000001000000000000100101000010001001 \\ 000000100000000000100010010111100111 \\ 000000010000000000100001101001000011 \\ 000000001000000000100001010100100001 \\ 000000000100000000101111000110101101 \\ 000000000010000000101100001001011000 \\ 000000000001000000100011000001111011 \\ 000000000000100000110110010001111000 \\ 000000000000010000000110101001111100 \\ 000000000000001000001100111001001110 \\ 000000000000000100000111110001011100 \\ 000000000000000010000110011111000000 \\ 000000000000000001011100011001000100 \end{bmatrix}$$

$$G_L^{27} = \begin{bmatrix}
1000000000000000001101110001110110010\\
0100000000000000001101001100000000011\\
0010000000000000001000000010110110010\\
0001000000000000001100101011001001001\\
0000100000000000001101101110001010110\\
0000010000000000001001010000100010011\\
0000001000000000001000100101111001111\\
0000000100000000001000011010010000011\\
0000000010000000001000000111000110100\\
0000000001000000001011000010010110011\\
0000000000100000001000110000011110000\\
0000000000010000000010101100111110\\
0000000000001000000011000111000111110\\
0000000000000100000011001110010011\\
0000000000000010000000111111000101111\\
0000000000000001000000011100111100000\\
0000000000000000011000001110110001
\end{bmatrix}$$

$$G_L^{28} = \begin{bmatrix}
1000000000000000000010010111010000010\\
0100000000000000000001101011010000010\\
0010000000000000000001111001000100010\\
0001000000000000000011001101101111010\\
0000100000000000000010001000101100110\\
0000010000000000000010111011111001010\\
0000001000000000000010000001010100111\\
0000000100000000000010011101101000010\\
0000000010000000000011100000000111111\\
0000000001000000001011000010010110011\\
0000000000100000001000110000011110011\\
0000000000010000000110100100010001100\\
0000000000001000000011100111100100111\\
0000000000000100000011000111100010111\\
0000000000000010000000111111000101111\\
0000000000000001000000011100111100000\\
0000000000000000011001001110110000001
\end{bmatrix}$$

$$G_L^{29} = \begin{bmatrix}
1000000000000000001001100101100000010\\
0100000000000000001010011001100000010\\
0010000000000000001011000011000000011\\
0001000000000000001000111111011111001\\
0000100000000000001001111010011100110\\
0000010000000000001001010000100010011\\
0000001000000000001000100101111001111\\
0000000100000000001000011010010000011\\
0000000010000000001000010101001000011\\
0000000001000000000000010101100111110\\
0000000000100000000011010010101011010\\
0000000000010000000110011010111001\\
0000000000001000000011010100010001100\\
0000000000000100000011000111100100111\\
0000000000000010000000111111000101111\\
0000000000000001000000011100111100000\\
0000000000000000011101010110000000001
\end{bmatrix}$$

$$G_L^{30} = \begin{bmatrix}
1000000000000000001000100011011000010\\
0100000000000000001011011111011000010\\
0010000000000000001011000011000000011\\
0001000000000000001001111001100111010\\
0000100000000000001000111100100100101\\
0000010000000000001001010000100010011\\
0000001000000000001000100101111001111\\
0000000100000000001000011010010000011\\
0000000010000000001000010101001000011\\
0000000001000000000000010100111011111101\\
0000000000100000000011110010010011010\\
0000000000010000000100010110101110010\\
0000000000001000000011010100010001111\\
0000000000000100000011000111100100111\\
0000000000000010000000111111000101000\\
0000000000000001000000011100111110000\\
0000000000000000011001001110110000001
\end{bmatrix}$$

$$G_L^{31} = \begin{bmatrix}
1000000000000000000110110011111010010\\
0100000000000000001010011111111010010\\
0010000000000000000011000000111010010\\
0001000000000000000111010010000101001\\
0000100000000000001101011000000110101\\
0000010000000000000001010011011000010\\
0000001000000000000001001110000011110\\
0000000100000000000000110011010100010\\
0000000010000000000000101101100100010\\
0000000001000000000011100000000111111\\
0000000000100000001011000010010110011\\
0000000000010000001000110000011110000\\
0000000000001000000110101000100011111\\
0000000000000100000011000111100100111\\
0000000000000010000000111111000101000\\
0000000000000001000000011100111110000\\
0000000000000000010000000111110101001
\end{bmatrix}$$

$$G_L^{32} = \begin{bmatrix}
1000000000000000000100101111110111010\\
0100000000000000000011010111101110010\\
0010000000000000000111100100110111010\\
0001000000000000000111000000101001010\\
0000100000000000001000010000010111110\\
0000010000000000001011101110101010010\\
0000001000000000001000000100011110110\\
0000000100000000001001111011001110010\\
0000000010000000001001100101111111010\\
0000000001000000000111000000000111100\\
0000000000100000001011000010010110011\\
0000000000010000001000110000011110000\\
0000000000001000000110101000100011111\\
0000000000000100000011000111100100111\\
0000000000000010000000111111000101000\\
0000000000000001000000011100111110011\\
0000000000000000011001001111101110011
\end{bmatrix}$$

$$G_L^{33} = \begin{bmatrix}
1000000000000000001010001101010000010\\
0100000000000000001001110001010000010\\
0010000000000000001011000011000000011\\
0001000000000000001011010111101111001\\
0000100000000000001010010010101100101\\
0000010000000000001001010000100010011\\
0000001000000000001000100101111001100\\
0000000100000000001000011010010000011\\
0000000010000000001000010101001000011\\
0000000001000000000011111101010111110\\
0000000000100000000010111000110110\\
0000000000010000000100101011111010\\
0000000000001000000110101000100011111\\
0000000000000100000101000100101100\\
0000000000000010000000111111000101111\\
0000000000000001000000011100111110011\\
0000000000000000010011110101000001
\end{bmatrix}$$

$$G_L^{34} = \begin{bmatrix}
1000000000000000000111011000000000011\\
0100000000000000001100011111011010010\\
0010000000000000001010010000011010010\\
0001000000000000001110111001100101010\\
0000100000000000001110101111111100100\\
0000010000000000001000000011111000010\\
0000001000000000001000100101111001111\\
0000000100000000001000011010010000011\\
0000000010000000001000010101001000011\\
0000000001000000000110010011011101101\\
0000000000100000000100110010010001001\\
0000000000010000000100001000011111011\\
0000000000001000000100001000111001011110\\
0000000000000100000101000100101100\\
0000000000000010000000111111000101111\\
0000000000000001000000011100111110011\\
0000000000000000010100111011010001
\end{bmatrix}$$

$$G_L^{35} = \begin{bmatrix}
100000000000000000101100010010100010\\
010000000000000000100011100010100010\\
001000000000000000101100001100000011\\
000100000000000000101001111010111010\\
000010000000000000101101101101010110\\
000001000000000000100101000010010011\\
000000100000000000100100101111001100\\
000000010000000000100001101001000011\\
000000001000000000101000101001000011\\
000000000100000000101101001011111101\\
000000000010000000000101011000110001\\
000000000001000000001101100100111010\\
000000000000100000011010100010001111\\
000000000000010000010100010010101111\\
000000000000001000001100111001001111\\
000000000000000100000001111100010100\\
000000000000000010000001100111110011\\
000000000000000000101110100101000001
\end{bmatrix}$$

$$G_L^{36} = \begin{bmatrix}
100000000000000000100001001100100110\\
010000000000000000101110111001000110\\
001000000000000000101100001100000011\\
000100000000000000100100100111000101\\
000010000000000000101001001001100101\\
000001000000000000101000011011101000\\
000000100000000000100101000001010011\\
000000010000000000100100101001000011\\
000000001000000000101110001100000011\\
000000000100000000101110011011101101\\
000000000010000000001100011101110010\\
000000000001000000001011100111001001\\
000000000000100000001010000111010100\\
000000000000010000001001010100011100\\
000000000000001000001100111101111011\\
000000000000000100000001111110000111\\
000000000000000010000001100011011100\\
000000000000000000100001101000011101
\end{bmatrix}$$

$$G_L^{37} = \begin{bmatrix}
100000000000000000001101111000111010\\
010000000000000000000100010001110110\\
001000000000000000011010110100111010\\
000100000000000000010000100110001110\\
000010000000000000011000001110010010\\
000001000000000000010100001101010110\\
000000100000000000010111111011001110\\
000000010000000000100111110011001110\\
000000001000000000101111010001100110\\
000000000100000000011100000011010100\\
000000000010000000010100010001011100\\
000000000001000000010010001001011111\\
000000000000100000010010011100000111\\
000000000000010000011100100011100011\\
000000000000000100010010010101011011\\
000000000000000010000001111010010111\\
000000000000000000110110111000111010 1
\end{bmatrix}$$

$$G_L^{38} = \begin{bmatrix}
100000000000000000001001101100110101 0\\
010000000000000000000110011010100110\\
001000000000000000011110100000110101 0\\
000100000000000000011000001110010001\\
000010000000000000010000100110000101\\
000001000000000000010101110110110110\\
000000100000000000010101010111011010\\
000000010000000000100111110000101010\\
000000001000000000111000000110100100\\
000000000100000000101010001000101100\\
000000000010000000100101000100101111\\
000000000001000000100100111000001111\\
000000000000100000100011100000111011\\
000000000000010000011100100011100011\\
000000000000001000100101001010110111\\
000000000000000100000011110100101111\\
000000000000000010000001100110011111
\end{bmatrix}$$

$$G_L^{39} = \begin{bmatrix}
100000000000000000000100111100100111 0\\
010000000000000000001011001100100111 0\\
001000000000000000010011110000100111 0\\
000100000000000000010101110110110110\\
000010000000000000010100001101010001\\
000001000000000000011000001110010001\\
000000100000000000011111110101011110\\
000000010000000000110101011110111110\\
000000001000000000111101010000001110\\
000000000100000000011100000110100100\\
000000000010000000101010001000101100\\
000000000001000000100101000100101111\\
000000000000100000100100111000001111\\
000000000000010000011100100011100011\\
000000000000001000100101001010110111\\
000000000000000100000011110100101111\\
000000000000000010000001111111111001001101
\end{bmatrix}$$

$$G_L^{40} = \begin{bmatrix}
100000000000000000010010000111010111 0\\
010000000000000000011101110111010111 0\\
001000000000000000001010010110101110\\
000100000000000000010101110110110101\\
000010000000000000011010110011001101\\
000001000000000000010101110110110110\\
000000100000000000011000011101110100\\
000000010000000000101110110011001000\\
000000001000000000010000111110001110\\
000000000100000000011100001111101111\\
000000000010000000011000100110100100\\
000000000001000000101100001101000111\\
000000000000100000101101111001110111\\
000000000000010000011100100111000011\\
000000000000001000001100001111100000\\
000000000000000100000011100111110111\\
000000000000000010000000111010111111\\
000000000000000001010010001110101101
\end{bmatrix}$$

$$G_L^{41} = \begin{bmatrix}
100000000000000000000100111110000001 0\\
010000000000000000001011001110000001 0\\
001000000000000000010011110010000001 0\\
000100000000000000011101010100000001 0\\
000010000000000000011110101000000001 0\\
000001000000000000011111110001111001 1\\
000000100000000000011111001101110101 1\\
000000010000000000111000000011100111\\
000000001000000000100101000011010111\\
000000000100000000100110001010110111\\
000000000010000000101011010010001 1\\
000000000001000000100100010011011111\\
000000000000100000010001101100101000\\
000000000000010000010000101101011000\\
000000000000001000001100001111010111\\
000000000000000100000011001100111111\\
000000000000000010000000111001111111\\
000000000000000001111111111100000001
\end{bmatrix}$$

Appendix B

Codes $\mathcal{D}$

This appendix lists the $(12, 6, 4)$ self-dual codes obtained from the initial 19 cases of a $(36, 18, 8)$ code $\mathcal{L}_i \in \mathbb{L}$ fixed by a fixed-point-free automorphism of order 3. By doubling the coordinates, these codes can be used to determine the 6-dimensional overlap between the particular code $\mathcal{L}_i$ and the desired permutation of $\mathcal{G}_{24}$ (see Chapter 4, Section 4.2.2). G_j^i denotes a generator matrix for code $\mathcal{D}_j^i$ that was obtained from $\mathcal{L}_i$ (listed in Appendix A reorganized for it to be fixed by the automorphism σ_j^i (listed in Chapter 4, Section 4.2.5).

$$G_1^1 = \begin{bmatrix} 101100000001 \\ 111000000010 \\ 101001010000 \\ 101010100000 \\ 011111000100 \\ 110111001000 \end{bmatrix} \quad G_1^3 = \begin{bmatrix} 111101100000 \\ 010110000001 \\ 001110000100 \\ 100110001000 \\ 111011000010 \\ 000111010000 \end{bmatrix} \quad G_1^6 = \begin{bmatrix} 011000001010 \\ 011000010001 \\ 111001000000 \\ 011110000000 \\ 110010011100 \\ 101010111000 \end{bmatrix}$$

$$G_1^7 = \begin{bmatrix} 001011001000 \\ 011001010000 \\ 110111100000 \\ 111110000001 \\ 101001000010 \\ 001101000100 \end{bmatrix} \quad G_1^8 = \begin{bmatrix} 111110001000 \\ 110001010000 \\ 100011100000 \\ 100101000001 \\ 101001000010 \\ 011111000100 \end{bmatrix} \quad G_1^9 = \begin{bmatrix} 110001001000 \\ 111110010000 \\ 011001100000 \\ 101111000001 \\ 010011000010 \\ 010101000100 \end{bmatrix}$$

$$G_1^{11} = \begin{bmatrix} 110011000000 \\ 011101011000 \\ 101101101000 \\ 110000001001 \\ 111000000010 \\ 110100000100 \end{bmatrix} \quad G_1^{12} = \begin{bmatrix} 001111000000 \\ 011001001000 \\ 101001100000 \\ 111010010001 \\ 001001010010 \\ 110011010100 \end{bmatrix} \quad G_1^{14} = \begin{bmatrix} 010111000000 \\ 110010001000 \\ 010010110000 \\ 111001010001 \\ 011010000010 \\ 101011010100 \end{bmatrix}$$

$$G_1^{16} = \begin{bmatrix} 011011000000 \\ 001101010000 \\ 110101101000 \\ 111100001001 \\ 101001000010 \\ 001001001100 \end{bmatrix} \quad G_1^{21} = \begin{bmatrix} 001111000000 \\ 011001001000 \\ 001001110000 \\ 101001000001 \\ 111010010010 \\ 110011010100 \end{bmatrix} \quad G_2^{21} = \begin{bmatrix} 001111000000 \\ 011001001000 \\ 001001110000 \\ 101001000001 \\ 111010010010 \\ 110011010100 \end{bmatrix}$$

$$G_1^{22} = \begin{bmatrix} 111100000000 \\ 011001010000 \\ 110011101000 \\ 101011001001 \\ 011000001010 \\ 011010000100 \end{bmatrix} \quad G_2^{22} = \begin{bmatrix} 110010001000 \\ 100011010000 \\ 101010100000 \\ 011111000001 \\ 111101000010 \\ 100110000100 \end{bmatrix} \quad G_3^{22} = \begin{bmatrix} 100111000000 \\ 101001001000 \\ 110001100000 \\ 111010010001 \\ 011011010010 \\ 100001010100 \end{bmatrix}$$

$$G_1^{25} = \begin{bmatrix} 111111000000 \\ 100010011000 \\ 001010101000 \\ 010010001001 \\ 111001001010 \\ 000011001100 \end{bmatrix} \quad G_2^{25} = \begin{bmatrix} 111111000000 \\ 100010011000 \\ 001010101000 \\ 010010001001 \\ 111001001010 \\ 000011001100 \end{bmatrix} \quad G_3^{25} = \begin{bmatrix} 110110000000 \\ 100011010000 \\ 011011101000 \\ 111001001001 \\ 100010001010 \\ 101010000100 \end{bmatrix}$$

$$G_4^{25} = \begin{bmatrix} 011100001000 \\ 110100010000 \\ 010110100000 \\ 010101000001 \\ 101111000010 \\ 111011000100 \end{bmatrix}$$

Appendix C

Codes $\mathcal{X}$

This appendix lists the 38 generator matrices for the codes $\mathcal{X}_i$, $1 \leq i \leq 36$, and the codes $\mathcal{X}_A$, $\mathcal{X}_B$.

$$G_1^{\times} = \left[\begin{array}{cccccccccccc}
100011 & 000000 & 000000 & 000000 & 000000 & 000000 & 101010 & 010101 & 010101 & 001001 & 100011 & 110001 \\
010010 & 000000 & 000000 & 000000 & 000000 & 000000 & 111111 & 111111 & 111111 & 101101 & 010010 & 001001 \\
001001 & 000000 & 000000 & 000000 & 000000 & 000000 & 111111 & 111111 & 111111 & 110110 & 001001 & 100100 \\
000111 & 000000 & 000000 & 000000 & 000000 & 000000 & 010101 & 101010 & 101010 & 010010 & 000111 & 100011 \\
000000 & 100011 & 000000 & 000000 & 000000 & 000000 & 101101 & 010010 & 011100 & 100011 & 010101 & 100100 \\
000000 & 010010 & 000000 & 000000 & 000000 & 000000 & 011011 & 011011 & 010010 & 010010 & 111111 & 110110 \\
000000 & 001001 & 000000 & 000000 & 000000 & 000000 & 101101 & 101101 & 001001 & 001001 & 111111 & 011011 \\
000000 & 000111 & 000000 & 000000 & 000000 & 000000 & 011011 & 100100 & 111000 & 000111 & 101010 & 001001 \\
000000 & 000000 & 100011 & 000000 & 000000 & 000000 & 001110 & 001110 & 010010 & 101010 & 111000 & 100011 \\
000000 & 000000 & 010010 & 000000 & 000000 & 000000 & 001001 & 001001 & 011011 & 111111 & 100100 & 010010 \\
000000 & 000000 & 001001 & 000000 & 000000 & 000000 & 100100 & 100100 & 101101 & 111111 & 010010 & 001001 \\
000000 & 000000 & 000111 & 000000 & 000000 & 000000 & 011100 & 011100 & 100100 & 010101 & 110001 & 000111 \\
000000 & 000000 & 000000 & 100011 & 000000 & 000000 & 001001 & 111111 & 010101 & 001110 & 011011 & 010101 \\
000000 & 000000 & 000000 & 010010 & 000000 & 000000 & 101101 & 000000 & 111111 & 001001 & 110110 & 111111 \\
000000 & 000000 & 000000 & 001001 & 000000 & 000000 & 110110 & 000000 & 111111 & 100100 & 011011 & 111111 \\
000000 & 000000 & 000000 & 000111 & 000000 & 000000 & 010010 & 111111 & 101010 & 011100 & 110110 & 101010 \\
000000 & 000000 & 000000 & 000000 & 100011 & 000000 & 010101 & 000000 & 100011 & 100011 & 001001 & 111111 \\
000000 & 000000 & 000000 & 000000 & 010010 & 000000 & 111111 & 000000 & 010010 & 010010 & 101101 & 000000 \\
000000 & 000000 & 000000 & 000000 & 001001 & 000000 & 111111 & 000000 & 001001 & 001001 & 110110 & 000000 \\
000000 & 000000 & 000000 & 000000 & 000111 & 000000 & 101010 & 000000 & 000111 & 000111 & 010010 & 111111 \\
000000 & 000000 & 000000 & 000000 & 000000 & 100011 & 111111 & 100011 & 101010 & 001110 & 100100 & 000000 \\
000000 & 000000 & 000000 & 000000 & 000000 & 010010 & 000000 & 010010 & 111111 & 001001 & 110110 & 000000 \\
000000 & 000000 & 000000 & 000000 & 000000 & 001001 & 000000 & 001001 & 111111 & 100100 & 011011 & 000000 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000111 & 111111 & 000111 & 010101 & 011100 & 001001 & 000000
\end{array}\right]$$

$$G_2^{\times} = \begin{bmatrix}
100011 & 000000 & 000000 & 000000 & 000000 & 000000 & 101010 & 010101 & 010101 & 001001 & 011100 & 001110 \\
010010 & 000000 & 000000 & 000000 & 000000 & 000000 & 111111 & 111111 & 111111 & 101101 & 010010 & 001001 \\
001001 & 000000 & 000000 & 000000 & 000000 & 000000 & 111111 & 111111 & 111111 & 110110 & 001001 & 100100 \\
000111 & 000000 & 000000 & 000000 & 000000 & 000000 & 010101 & 101010 & 101010 & 010010 & 111000 & 011100 \\
000000 & 100011 & 000000 & 000000 & 000000 & 000000 & 010010 & 010010 & 100011 & 100011 & 010101 & 011011 \\
000000 & 010010 & 000000 & 000000 & 000000 & 000000 & 011011 & 011011 & 010010 & 010010 & 111111 & 110110 \\
000000 & 001001 & 000000 & 000000 & 000000 & 000000 & 101101 & 101101 & 001001 & 001001 & 111111 & 011011 \\
000000 & 000111 & 000000 & 000000 & 000000 & 000000 & 100100 & 100100 & 000111 & 000111 & 101010 & 110110 \\
000000 & 000000 & 100011 & 000000 & 000000 & 000000 & 001110 & 001110 & 010010 & 101010 & 111000 & 100011 \\
000000 & 000000 & 010010 & 000000 & 000000 & 000000 & 001001 & 001001 & 011011 & 111111 & 100100 & 010010 \\
000000 & 000000 & 001001 & 000000 & 000000 & 000000 & 100100 & 100100 & 101101 & 111111 & 010010 & 001001 \\
000000 & 000000 & 000111 & 000000 & 000000 & 000000 & 011100 & 011100 & 100100 & 010101 & 110001 & 000111 \\
000000 & 000000 & 000000 & 100011 & 000000 & 000000 & 110110 & 111111 & 101010 & 001110 & 100100 & 010101 \\
000000 & 000000 & 000000 & 010010 & 000000 & 000000 & 101101 & 000000 & 111111 & 001001 & 110110 & 111111 \\
000000 & 000000 & 000000 & 001001 & 000000 & 000000 & 110110 & 000000 & 111111 & 100100 & 011011 & 111111 \\
000000 & 000000 & 000000 & 000111 & 000000 & 000000 & 101101 & 111111 & 010101 & 011100 & 001001 & 101010 \\
000000 & 000000 & 000000 & 000000 & 100011 & 000000 & 010101 & 000000 & 100011 & 100011 & 001001 & 111111 \\
000000 & 000000 & 000000 & 000000 & 010010 & 000000 & 111111 & 000000 & 010010 & 010010 & 101101 & 000000 \\
000000 & 000000 & 000000 & 000000 & 001001 & 000000 & 111111 & 000000 & 001001 & 001001 & 110110 & 000000 \\
000000 & 000000 & 000000 & 000000 & 000111 & 000000 & 101010 & 000000 & 000111 & 000111 & 010010 & 111111 \\
000000 & 000000 & 000000 & 000000 & 000000 & 100011 & 111111 & 100011 & 101010 & 001110 & 011011 & 111111 \\
000000 & 000000 & 000000 & 000000 & 000000 & 010010 & 000000 & 010010 & 111111 & 001001 & 110110 & 000000 \\
000000 & 000000 & 000000 & 000000 & 000000 & 001001 & 000000 & 001001 & 111111 & 100100 & 011011 & 000000 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000111 & 111111 & 000111 & 010101 & 011100 & 110110 & 111111
\end{bmatrix}$$

$$G_3^{\times} = \begin{bmatrix}
100011 & 000000 & 000000 & 000000 & 000000 & 001110 & 001001 & 101010 & 001110 & 101101 & 110110 & 111111 \\
010010 & 000000 & 000000 & 000000 & 000000 & 001001 & 000000 & 111111 & 001001 & 110110 & 110110 & 011011 \\
001001 & 000000 & 000000 & 000000 & 000000 & 001110 & 001110 & 101010 & 011011 & 111111 & 000111 & 011011 \\
000111 & 000000 & 000000 & 000000 & 000000 & 001001 & 001110 & 111111 & 100011 & 010010 & 010101 & 010010 \\
000000 & 100011 & 000000 & 000000 & 000000 & 001110 & 001001 & 001110 & 000000 & 010010 & 110110 & 010101 \\
000000 & 010010 & 000000 & 000000 & 000000 & 001001 & 000000 & 001001 & 000000 & 110110 & 110110 & 100100 \\
000000 & 001001 & 000000 & 000000 & 000000 & 001110 & 001110 & 110001 & 111111 & 111111 & 000111 & 100100 \\
000000 & 000111 & 000000 & 000000 & 000000 & 001001 & 001110 & 110110 & 111111 & 101101 & 010101 & 000111 \\
000000 & 000000 & 100011 & 000000 & 000000 & 000111 & 000000 & 011100 & 001001 & 011100 & 101101 & 111111 \\
000000 & 000000 & 010010 & 000000 & 000000 & 001110 & 001110 & 000111 & 010010 & 110110 & 000111 & 110110 \\
000000 & 000000 & 001001 & 000000 & 000000 & 000111 & 000111 & 100011 & 001001 & 011011 & 100011 & 011011 \\
000000 & 000000 & 000111 & 000000 & 000000 & 001110 & 000000 & 111000 & 010010 & 111000 & 011011 & 111111 \\
000000 & 000000 & 000000 & 100011 & 000000 & 001001 & 001110 & 000111 & 000111 & 001110 & 000000 & 010101 \\
000000 & 000000 & 000000 & 010010 & 000000 & 000111 & 000111 & 110001 & 011011 & 101101 & 011100 & 001001 \\
000000 & 000000 & 000000 & 001001 & 000000 & 001001 & 000000 & 101101 & 010010 & 111111 & 001001 & 001001 \\
000000 & 000000 & 000000 & 000111 & 000000 & 000111 & 000000 & 100100 & 110001 & 010101 & 111000 & 000111 \\
000000 & 000000 & 000000 & 000000 & 100011 & 000111 & 000111 & 111111 & 010101 & 001110 & 011011 & 100011 \\
000000 & 000000 & 000000 & 000000 & 010010 & 001110 & 000111 & 010101 & 000000 & 000000 & 110001 & 111111 \\
000000 & 000000 & 000000 & 000000 & 001001 & 000111 & 001110 & 101010 & 000000 & 101101 & 100011 & 100100 \\
000000 & 000000 & 000000 & 000000 & 000111 & 001110 & 001110 & 111111 & 101010 & 011100 & 110110 & 000111 \\
000000 & 000000 & 000000 & 000000 & 000000 & 101010 & 001110 & 010101 & 111111 & 100100 & 011100 & 110110 \\
000000 & 000000 & 000000 & 000000 & 000000 & 010101 & 000111 & 101010 & 111111 & 010010 & 001110 & 011011 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 101101 & 000000 & 000000 & 101101 & 011011 & 011011 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 011011 & 000000 & 000000 & 011011 & 110110 & 110110
\end{bmatrix}$$

$$G_4^{\times} = \begin{bmatrix}
100011 & 000000 & 000000 & 000000 & 000000 & 001110 & 001001 & 010101 & 110001 & 010010 & 001001 & 111111 \\
010010 & 000000 & 000000 & 000000 & 000000 & 001001 & 000000 & 111111 & 001001 & 110110 & 110110 & 011011 \\
001001 & 000000 & 000000 & 000000 & 000000 & 001110 & 001110 & 101010 & 011011 & 000000 & 000111 & 100100 \\
000111 & 000000 & 000000 & 000000 & 000000 & 001001 & 001110 & 000000 & 011100 & 010010 & 101010 & 101101 \\
000000 & 100011 & 000000 & 000000 & 000000 & 001110 & 001001 & 110001 & 000000 & 010010 & 110110 & 010101 \\
000000 & 010010 & 000000 & 000000 & 000000 & 001001 & 000000 & 001001 & 000000 & 110110 & 110110 & 100100 \\
000000 & 001001 & 000000 & 000000 & 000000 & 001110 & 001110 & 110001 & 111111 & 000000 & 000111 & 011011 \\
000000 & 000111 & 000000 & 000000 & 000000 & 001001 & 001110 & 001001 & 111111 & 010010 & 010101 & 111000 \\
000000 & 000000 & 100011 & 000000 & 000000 & 000111 & 000000 & 011100 & 001001 & 100011 & 010010 & 000000 \\
000000 & 000000 & 010010 & 000000 & 000000 & 001110 & 001110 & 000111 & 010010 & 001001 & 000111 & 001001 \\
000000 & 000000 & 001001 & 000000 & 000000 & 000111 & 000111 & 100011 & 001001 & 100100 & 100011 & 100100 \\
000000 & 000000 & 000111 & 000000 & 000000 & 001110 & 000000 & 111000 & 010010 & 000111 & 100100 & 000000 \\
000000 & 000000 & 000000 & 100011 & 000000 & 001001 & 001110 & 000111 & 111000 & 001110 & 000000 & 101010 \\
000000 & 000000 & 000000 & 010010 & 000000 & 000111 & 000111 & 110001 & 011011 & 010010 & 011100 & 110110 \\
000000 & 000000 & 000000 & 001001 & 000000 & 001001 & 000000 & 101101 & 010010 & 111111 & 001001 & 001001 \\
000000 & 000000 & 000000 & 000111 & 000000 & 000111 & 000000 & 100100 & 001110 & 101010 & 111000 & 000111 \\
000000 & 000000 & 000000 & 000000 & 100011 & 000111 & 000111 & 111111 & 010101 & 001110 & 011011 & 100011 \\
000000 & 000000 & 000000 & 000000 & 010010 & 001110 & 000111 & 010101 & 000000 & 111111 & 110001 & 000000 \\
000000 & 000000 & 000000 & 000000 & 001001 & 000111 & 001110 & 101010 & 000000 & 010010 & 100011 & 011011 \\
000000 & 000000 & 000000 & 000000 & 000111 & 001110 & 001110 & 111111 & 101010 & 011100 & 110110 & 000111 \\
000000 & 000000 & 000000 & 000000 & 000000 & 101010 & 001110 & 010101 & 111111 & 011011 & 011100 & 001001 \\
000000 & 000000 & 000000 & 000000 & 000000 & 010101 & 000111 & 101010 & 111111 & 101101 & 001110 & 100100 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 101101 & 000000 & 000000 & 101101 & 011011 & 011011 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 011011 & 000000 & 000000 & 011011 & 110110 & 110110
\end{bmatrix}$$

$$G_5^{\times} = \begin{bmatrix}
100011 & 000000 & 000000 & 000000 & 000000 & 010101 & 000111 & 000000 & 111111 & 111000 & 110110 & 111111 \\
010010 & 000000 & 000000 & 000000 & 000000 & 000000 & 011011 & 000111 & 010101 & 111111 & 010101 & 110001 \\
001001 & 000000 & 000000 & 000000 & 000000 & 000000 & 010010 & 000111 & 101010 & 110110 & 110001 & 110001 \\
000111 & 000000 & 000000 & 000000 & 000000 & 010101 & 001110 & 000111 & 010101 & 010101 & 101010 & 001110 \\
000000 & 100011 & 000000 & 000000 & 000111 & 000111 & 010101 & 000111 & 100011 & 001001 & 011011 & 011011 \\
000000 & 010010 & 000000 & 000000 & 000000 & 010010 & 000111 & 000000 & 111000 & 111000 & 101010 & 101101 \\
000000 & 001001 & 000000 & 000000 & 000000 & 001001 & 011100 & 000000 & 100011 & 100011 & 101010 & 110110 \\
000000 & 000111 & 000000 & 000000 & 000111 & 011100 & 011011 & 000000 & 111000 & 011100 & 001001 & 110001 \\
000000 & 000000 & 100011 & 000000 & 000111 & 000111 & 000111 & 000111 & 001001 & 111111 & 100011 & 001001 \\
000000 & 000000 & 010010 & 000000 & 000000 & 010010 & 011100 & 000000 & 000111 & 010101 & 001110 & 110110 \\
000000 & 000000 & 001001 & 000000 & 000000 & 001001 & 001110 & 000000 & 100011 & 101010 & 000111 & 011011 \\
000000 & 000000 & 000111 & 000000 & 000111 & 011100 & 000000 & 000000 & 010010 & 001110 & 000111 & 010101 \\
000000 & 000000 & 100011 & 000000 & 000111 & 011011 & 001110 & 000111 & 011100 & 011100 & 100011 & 101010 \\
000000 & 000000 & 000000 & 100011 & 000111 & 011011 & 001110 & 000111 & 011100 & 011100 & 100011 & 101010 \\
000000 & 000000 & 000000 & 010010 & 000000 & 000000 & 001110 & 000000 & 000111 & 111000 & 110001 & 100100 \\
000000 & 000000 & 000000 & 001001 & 000000 & 000000 & 000111 & 000000 & 100011 & 011100 & 111000 & 010010 \\
000000 & 000000 & 000000 & 000111 & 000111 & 011011 & 010010 & 000111 & 010010 & 101101 & 000000 & 100011 \\
000000 & 000000 & 000000 & 000000 & 100100 & 001001 & 000111 & 000111 & 000000 & 111000 & 000000 & 100011 \\
000000 & 000000 & 000000 & 000000 & 010010 & 011011 & 011100 & 000000 & 101010 & 110001 & 011100 & 001001 \\
000000 & 000000 & 000000 & 000000 & 001001 & 010010 & 001110 & 000111 & 111111 & 011100 & 001001 & 010101 \\
000000 & 000000 & 000000 & 000000 & 000000 & 111111 & 000000 & 000111 & 101010 & 100100 & 000111 & 110001 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 111111 & 000000 & 111111 & 111111 & 111111 & 000000 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 100100 & 111111 & 110110 & 100100 & 001001 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 010010 & 111111 & 011011 & 010010 & 100100 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 001001 & 111111 & 101101 & 001001 & 010010
\end{bmatrix}$$

$$G_6^{\times} = \begin{bmatrix}
100011 & 000000 & 000000 & 000000 & 000000 & 010101 & 010010 & 000000 & 010101 & 101101 & 011100 & 111111 \\
010010 & 000000 & 000000 & 000000 & 000000 & 000000 & 001110 & 000111 & 111111 & 010101 & 000000 & 001110 \\
001001 & 000000 & 000000 & 000000 & 000000 & 000000 & 000111 & 000111 & 000000 & 011100 & 100100 & 001110 \\
000111 & 000000 & 000000 & 000000 & 000000 & 010101 & 001110 & 000111 & 010101 & 101010 & 010101 & 110001 \\
000000 & 100011 & 000000 & 000000 & 000111 & 000111 & 000000 & 000111 & 110110 & 011100 & 001110 & 100100 \\
000000 & 010010 & 000000 & 000000 & 000000 & 010010 & 000111 & 000000 & 111000 & 111000 & 101010 & 101101 \\
000000 & 001001 & 000000 & 000000 & 000000 & 001001 & 011100 & 000000 & 100011 & 100011 & 101010 & 110110 \\
000000 & 000111 & 000000 & 000000 & 000111 & 011100 & 011011 & 000000 & 000111 & 100011 & 001001 & 110001 \\
000000 & 000000 & 100011 & 000000 & 000111 & 000111 & 000111 & 000111 & 110110 & 111111 & 011100 & 110110 \\
000000 & 000000 & 010010 & 000000 & 000000 & 010010 & 011100 & 000000 & 000111 & 010101 & 001110 & 110110 \\
000000 & 000000 & 001001 & 000000 & 000000 & 001001 & 001110 & 000000 & 100011 & 101010 & 000111 & 011011 \\
000000 & 000000 & 000111 & 000000 & 000111 & 011100 & 010101 & 000000 & 000111 & 100100 & 101101 & 010101 \\
000000 & 000000 & 000000 & 100011 & 000111 & 011011 & 011011 & 000111 & 001001 & 001001 & 110110 & 101010 \\
000000 & 000000 & 000000 & 010010 & 000000 & 000000 & 001110 & 000000 & 000111 & 111000 & 110001 & 100100 \\
000000 & 000000 & 000000 & 001001 & 000000 & 000000 & 000111 & 000000 & 100011 & 011100 & 111000 & 010010 \\
000000 & 000000 & 000000 & 000111 & 000111 & 011011 & 000111 & 000111 & 000111 & 111000 & 010101 & 100011 \\
000000 & 000000 & 000000 & 000000 & 100100 & 001001 & 010010 & 000111 & 101010 & 010010 & 010101 & 011100 \\
000000 & 000000 & 000000 & 000000 & 010010 & 011011 & 011100 & 000000 & 101010 & 110001 & 011100 & 001001 \\
000000 & 000000 & 000000 & 000000 & 001001 & 010010 & 011011 & 000111 & 010101 & 110110 & 011100 & 101010 \\
000000 & 000000 & 000000 & 000000 & 000000 & 111111 & 010101 & 000111 & 000000 & 001110 & 010010 & 001110 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 111111 & 000000 & 111111 & 111111 & 111111 & 000000 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 100100 & 111111 & 110110 & 100100 & 001001 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 010010 & 111111 & 011011 & 010010 & 100100 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 001001 & 111111 & 101101 & 001001 & 010010
\end{bmatrix}$$

$$G_7^{\times} = \begin{bmatrix}
100011 & 000000 & 000000 & 000000 & 000000 & 010101 & 010010 & 000111 & 111111 & 110110 & 011011 & 001110 \\
010010 & 000000 & 000000 & 000000 & 000000 & 000000 & 001110 & 000000 & 101010 & 001110 & 000111 & 111111 \\
001001 & 000000 & 000000 & 000000 & 000000 & 000000 & 000111 & 000000 & 010101 & 000111 & 100011 & 111111 \\
000111 & 000000 & 000000 & 000000 & 000000 & 010101 & 001110 & 000111 & 101010 & 101010 & 010101 & 110001 \\
000000 & 100011 & 000000 & 000000 & 000111 & 000111 & 000000 & 000111 & 110110 & 011100 & 001110 & 100100 \\
000000 & 010010 & 000000 & 000000 & 000000 & 010010 & 010010 & 000111 & 000111 & 001001 & 111000 & 100011 \\
000000 & 001001 & 000000 & 000000 & 000000 & 001001 & 001001 & 000111 & 011100 & 010010 & 111000 & 111000 \\
000000 & 000111 & 000000 & 000000 & 000111 & 011100 & 001110 & 000000 & 101101 & 001001 & 011100 & 001110 \\
000000 & 000000 & 100011 & 000000 & 000111 & 000111 & 000111 & 000000 & 100011 & 100100 & 011011 & 000111 \\
000000 & 000000 & 010010 & 000000 & 000000 & 010010 & 001001 & 000111 & 111000 & 100100 & 011100 & 111000 \\
000000 & 000000 & 001001 & 000000 & 000000 & 001001 & 011011 & 000111 & 011100 & 011011 & 010101 & 010101 \\
000000 & 000000 & 000111 & 000000 & 000111 & 011100 & 000000 & 000111 & 111000 & 010101 & 111111 & 011011 \\
000000 & 000000 & 000000 & 100011 & 000111 & 011011 & 001110 & 000111 & 011100 & 100011 & 100011 & 010101 \\
000000 & 000000 & 000000 & 010010 & 000000 & 000000 & 011011 & 000111 & 111000 & 001001 & 100011 & 101010 \\
000000 & 000000 & 000000 & 001001 & 000000 & 000000 & 010010 & 000111 & 011100 & 101101 & 101010 & 011100 \\
000000 & 000000 & 000000 & 000111 & 000111 & 011011 & 000111 & 000000 & 101101 & 100011 & 010010 & 010010 \\
000000 & 000000 & 000000 & 000000 & 100100 & 001001 & 000111 & 000111 & 000000 & 111000 & 000000 & 100011 \\
000000 & 000000 & 000000 & 000000 & 010010 & 011011 & 001001 & 000111 & 010101 & 000000 & 001110 & 000111 \\
000000 & 000000 & 000000 & 000000 & 001001 & 010010 & 001110 & 000111 & 111111 & 011100 & 001001 & 010101 \\
000000 & 000000 & 000000 & 000000 & 000000 & 111111 & 010101 & 000000 & 010101 & 010101 & 010101 & 111111 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 111111 & 000000 & 111111 & 111111 & 111111 & 000000 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 100100 & 111111 & 110110 & 100100 & 001001 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 010010 & 111111 & 011011 & 010010 & 100100 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 001001 & 111111 & 101101 & 001001 & 010010
\end{bmatrix}$$

$$G_8^{\times} = \begin{bmatrix}
100011 & 000000 & 000000 & 000000 & 000000 & 000000 & 000111 & 100100 & 101101 & 001001 & 000111 & 010101 \\
010010 & 000000 & 000000 & 000000 & 000000 & 000000 & 100100 & 110110 & 011011 & 101101 & 100100 & 111111 \\
001001 & 000000 & 000000 & 000000 & 000000 & 000000 & 010010 & 011011 & 101101 & 110110 & 010010 & 111111 \\
000111 & 000000 & 000000 & 000000 & 000000 & 000000 & 001110 & 001001 & 011011 & 010010 & 001110 & 101010 \\
000000 & 100011 & 000000 & 000000 & 000000 & 000000 & 110001 & 100011 & 001001 & 000000 & 001001 & 100011 \\
000000 & 010010 & 000000 & 000000 & 000000 & 000000 & 001001 & 010010 & 101101 & 000000 & 101101 & 010010 \\
000000 & 001001 & 000000 & 000000 & 000000 & 000000 & 100100 & 001001 & 110110 & 000000 & 110110 & 001001 \\
000000 & 000111 & 000000 & 000000 & 000000 & 000000 & 100011 & 000111 & 010010 & 000000 & 010010 & 000111 \\
000000 & 000000 & 100011 & 000000 & 000000 & 000000 & 010010 & 010101 & 010101 & 010101 & 000111 & 011100 \\
000000 & 000000 & 010010 & 000000 & 000000 & 000000 & 011011 & 111111 & 111111 & 111111 & 100100 & 010010 \\
000000 & 000000 & 001001 & 000000 & 000000 & 000000 & 101101 & 111111 & 111111 & 111111 & 010010 & 001001 \\
000000 & 000000 & 000111 & 000000 & 000000 & 000000 & 100100 & 101010 & 101010 & 101010 & 001110 & 111000 \\
000000 & 000000 & 000000 & 100011 & 000000 & 000000 & 110001 & 001001 & 010010 & 011100 & 000000 & 000111 \\
000000 & 000000 & 000000 & 010010 & 000000 & 000000 & 001001 & 101101 & 011011 & 010010 & 000000 & 100100 \\
000000 & 000000 & 000000 & 001001 & 000000 & 000000 & 100100 & 110110 & 101101 & 001001 & 000000 & 010010 \\
000000 & 000000 & 000000 & 000111 & 000000 & 000000 & 100011 & 010010 & 100100 & 111000 & 000000 & 001110 \\
000000 & 000000 & 000000 & 000000 & 100011 & 000000 & 010101 & 110110 & 010101 & 111111 & 100100 & 111000 \\
000000 & 000000 & 000000 & 000000 & 010010 & 000000 & 111111 & 101101 & 111111 & 000000 & 110110 & 100100 \\
000000 & 000000 & 000000 & 000000 & 001001 & 000000 & 111111 & 110110 & 111111 & 000000 & 011011 & 010010 \\
000000 & 000000 & 000000 & 000000 & 000111 & 000000 & 101010 & 101101 & 101010 & 111111 & 001001 & 110001 \\
000000 & 000000 & 000000 & 000000 & 000000 & 100011 & 100011 & 111000 & 101010 & 011100 & 000111 & 010010 \\
000000 & 000000 & 000000 & 000000 & 000000 & 010010 & 010010 & 100100 & 111111 & 010010 & 100100 & 011011 \\
000000 & 000000 & 000000 & 000000 & 000000 & 001001 & 001001 & 010010 & 111111 & 001001 & 010010 & 101101 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000111 & 000111 & 110001 & 010101 & 111000 & 001110 & 100100
\end{bmatrix}$$

$$G_9^{\times} = \begin{bmatrix}
100011 & 000000 & 000000 & 000000 & 000000 & 000000 & 000111 & 100100 & 010010 & 110110 & 000111 & 101010 \\
010010 & 000000 & 000000 & 000000 & 000000 & 000000 & 100100 & 110110 & 011011 & 101101 & 100100 & 111111 \\
001001 & 000000 & 000000 & 000000 & 000000 & 000000 & 010010 & 011011 & 101101 & 110110 & 010010 & 111111 \\
000111 & 000000 & 000000 & 000000 & 000000 & 000000 & 001110 & 001001 & 100100 & 101101 & 001110 & 010101 \\
000000 & 100011 & 000000 & 000000 & 000000 & 000000 & 110001 & 100011 & 001001 & 111111 & 110110 & 100011 \\
000000 & 010010 & 000000 & 000000 & 000000 & 000000 & 001001 & 010010 & 101101 & 000000 & 101101 & 010010 \\
000000 & 001001 & 000000 & 000000 & 000000 & 000000 & 100100 & 001001 & 110110 & 000000 & 110110 & 001001 \\
000000 & 000111 & 000000 & 000000 & 000000 & 000000 & 100011 & 000111 & 010010 & 111111 & 101101 & 000111 \\
000000 & 000000 & 100011 & 000000 & 000000 & 000000 & 010010 & 010101 & 010101 & 101010 & 111000 & 011100 \\
000000 & 000000 & 010010 & 000000 & 000000 & 000000 & 011011 & 111111 & 111111 & 111111 & 100100 & 010010 \\
000000 & 000000 & 001001 & 000000 & 000000 & 000000 & 101101 & 111111 & 111111 & 111111 & 010010 & 001001 \\
000000 & 000000 & 000111 & 000000 & 000000 & 000000 & 100100 & 101010 & 101010 & 010101 & 110001 & 111000 \\
000000 & 000000 & 000000 & 100011 & 000000 & 000000 & 110001 & 001001 & 101101 & 011100 & 111111 & 111000 \\
000000 & 000000 & 000000 & 010010 & 000000 & 000000 & 001001 & 101101 & 011011 & 010010 & 000000 & 100100 \\
000000 & 000000 & 000000 & 001001 & 000000 & 000000 & 100100 & 110110 & 101101 & 001001 & 000000 & 010010 \\
000000 & 000000 & 000000 & 000111 & 000000 & 000000 & 100011 & 010010 & 011011 & 111000 & 111111 & 110001 \\
000000 & 000000 & 000000 & 000000 & 100011 & 000000 & 010101 & 110110 & 010101 & 111111 & 100100 & 111000 \\
000000 & 000000 & 000000 & 000000 & 010010 & 000000 & 111111 & 101101 & 111111 & 000000 & 110110 & 100100 \\
000000 & 000000 & 000000 & 000000 & 001001 & 000000 & 111111 & 110110 & 111111 & 000000 & 011011 & 010010 \\
000000 & 000000 & 000000 & 000000 & 000111 & 000000 & 101010 & 101101 & 101010 & 111111 & 001001 & 110001 \\
000000 & 000000 & 000000 & 000000 & 000000 & 100011 & 100011 & 111000 & 101010 & 011100 & 000111 & 010010 \\
000000 & 000000 & 000000 & 000000 & 000000 & 010010 & 010010 & 100100 & 111111 & 010010 & 100100 & 011011 \\
000000 & 000000 & 000000 & 000000 & 000000 & 001001 & 001001 & 010010 & 111111 & 001001 & 010010 & 101101 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000111 & 000111 & 110001 & 010101 & 111000 & 001110 & 100100
\end{bmatrix}$$

$$G_{10}^{\times} = \begin{bmatrix}
100011 & 000000 & 000000 & 000000 & 000000 & 000000 & 110001 & 100011 & 001110 & 001001 & 100011 & 101010 \\
010010 & 000000 & 000000 & 000000 & 000000 & 000000 & 001001 & 010010 & 001001 & 101101 & 010010 & 111111 \\
001001 & 000000 & 000000 & 000000 & 000000 & 000000 & 100100 & 001001 & 100100 & 110110 & 001001 & 111111 \\
000111 & 000000 & 000000 & 000000 & 000000 & 000000 & 100011 & 000111 & 011100 & 010010 & 000111 & 010101 \\
000000 & 100011 & 000000 & 000000 & 000000 & 000000 & 100100 & 111000 & 110001 & 111000 & 000000 & 010010 \\
000000 & 010010 & 000000 & 000000 & 000000 & 000000 & 110110 & 100100 & 001001 & 100100 & 000000 & 011011 \\
000000 & 001001 & 000000 & 000000 & 000000 & 000000 & 011011 & 010010 & 100100 & 010010 & 000000 & 101101 \\
000000 & 000111 & 000000 & 000000 & 000000 & 000000 & 001001 & 110001 & 100011 & 110001 & 000000 & 100100 \\
000000 & 000000 & 100011 & 000000 & 000000 & 000000 & 010010 & 001001 & 001110 & 010101 & 110001 & 101101 \\
000000 & 000000 & 010010 & 000000 & 000000 & 000000 & 011011 & 101101 & 001001 & 111111 & 001001 & 011011 \\
000000 & 000000 & 001001 & 000000 & 000000 & 000000 & 101101 & 110110 & 100100 & 111111 & 100100 & 101101 \\
000000 & 000000 & 000111 & 000000 & 000000 & 000000 & 100100 & 010010 & 011100 & 101010 & 100011 & 011011 \\
000000 & 000000 & 000000 & 100011 & 000000 & 000000 & 000000 & 001001 & 110001 & 111000 & 010010 & 011100 \\
000000 & 000000 & 000000 & 010010 & 000000 & 000000 & 000000 & 101101 & 001001 & 100100 & 011011 & 010010 \\
000000 & 000000 & 000000 & 001001 & 000000 & 000000 & 000000 & 110110 & 100100 & 010010 & 101101 & 001001 \\
000000 & 000000 & 000000 & 000111 & 000000 & 000000 & 000000 & 010010 & 100011 & 110001 & 100100 & 111000 \\
000000 & 000000 & 000000 & 000000 & 100011 & 000000 & 100011 & 000000 & 010101 & 111000 & 111111 & 001001 \\
000000 & 000000 & 000000 & 000000 & 010010 & 000000 & 010010 & 000000 & 111111 & 100100 & 000000 & 101101 \\
000000 & 000000 & 000000 & 000000 & 001001 & 000000 & 001001 & 000000 & 111111 & 010010 & 000000 & 110110 \\
000000 & 000000 & 000000 & 000000 & 000111 & 000000 & 000111 & 000000 & 101010 & 110001 & 111111 & 010010 \\
000000 & 000000 & 000000 & 000000 & 000000 & 100011 & 111000 & 001110 & 100100 & 011100 & 101010 & 011100 \\
000000 & 000000 & 000000 & 000000 & 000000 & 010010 & 100100 & 001001 & 110110 & 010010 & 111111 & 010010 \\
000000 & 000000 & 000000 & 000000 & 000000 & 001001 & 010010 & 100100 & 011011 & 001001 & 111111 & 001001 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000111 & 110001 & 011100 & 001001 & 111000 & 010101 & 111000
\end{bmatrix}$$

$$G_{11}^{\times} = \begin{bmatrix}
100011 & 000000 & 000000 & 000000 & 000000 & 000000 & 110001 & 100011 & 001110 & 001001 & 011100 & 010101 \\
010010 & 000000 & 000000 & 000000 & 000000 & 000000 & 001001 & 010010 & 001001 & 101101 & 010010 & 111111 \\
001001 & 000000 & 000000 & 000000 & 000000 & 000000 & 100100 & 001001 & 100100 & 110110 & 001001 & 111111 \\
000111 & 000000 & 000000 & 000000 & 000000 & 000000 & 100011 & 000111 & 011100 & 010010 & 111000 & 101010 \\
000000 & 100011 & 000000 & 000000 & 000000 & 000000 & 100100 & 111000 & 110001 & 111000 & 111111 & 101101 \\
000000 & 010010 & 000000 & 000000 & 000000 & 000000 & 110110 & 100100 & 001001 & 100100 & 000000 & 011011 \\
000000 & 001001 & 000000 & 000000 & 000000 & 000000 & 011011 & 010010 & 100100 & 010010 & 000000 & 101101 \\
000000 & 000111 & 000000 & 000000 & 000000 & 000000 & 001001 & 110001 & 100011 & 110001 & 111111 & 011011 \\
000000 & 000000 & 100011 & 000000 & 000000 & 000000 & 101101 & 001001 & 110001 & 010101 & 110001 & 010010 \\
000000 & 000000 & 010010 & 000000 & 000000 & 000000 & 011011 & 101101 & 001001 & 111111 & 001001 & 011011 \\
000000 & 000000 & 001001 & 000000 & 000000 & 000000 & 101101 & 110110 & 100100 & 111111 & 100100 & 101101 \\
000000 & 000000 & 000111 & 000000 & 000000 & 000000 & 011011 & 010010 & 100011 & 101010 & 100011 & 100100 \\
000000 & 000000 & 000000 & 100011 & 000000 & 000000 & 111111 & 001001 & 001110 & 111000 & 101101 & 011100 \\
000000 & 000000 & 000000 & 010010 & 000000 & 000000 & 000000 & 101101 & 001001 & 100100 & 011011 & 010010 \\
000000 & 000000 & 000000 & 001001 & 000000 & 000000 & 000000 & 110110 & 100100 & 010010 & 101101 & 001001 \\
000000 & 000000 & 000000 & 000111 & 000000 & 000000 & 111111 & 010010 & 011100 & 110001 & 011011 & 111000 \\
000000 & 000000 & 000000 & 000000 & 100011 & 000000 & 100011 & 000000 & 010101 & 111000 & 111111 & 001001 \\
000000 & 000000 & 000000 & 000000 & 010010 & 000000 & 010010 & 000000 & 111111 & 100100 & 000000 & 101101 \\
000000 & 000000 & 000000 & 000000 & 001001 & 000000 & 001001 & 000000 & 111111 & 010010 & 000000 & 110110 \\
000000 & 000000 & 000000 & 000000 & 000111 & 000000 & 000111 & 000000 & 101010 & 110001 & 111111 & 010010 \\
000000 & 000000 & 000000 & 000000 & 000000 & 100011 & 111000 & 001110 & 100100 & 011100 & 101010 & 011100 \\
000000 & 000000 & 000000 & 000000 & 000000 & 010010 & 100100 & 001001 & 110110 & 010010 & 111111 & 010010 \\
000000 & 000000 & 000000 & 000000 & 000000 & 001001 & 010010 & 100100 & 011011 & 001001 & 111111 & 001001 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000111 & 110001 & 011100 & 001001 & 111000 & 010101 & 111000
\end{bmatrix}$$

$$G_{12}^{\times} = \begin{bmatrix}
100011 & 000000 & 000000 & 000000 & 000000 & 000000 & 110001 & 011100 & 110001 & 001001 & 011100 & 010101 \\
010010 & 000000 & 000000 & 000000 & 000000 & 000000 & 001001 & 010010 & 001001 & 101101 & 010010 & 111111 \\
001001 & 000000 & 000000 & 000000 & 000000 & 000000 & 100100 & 001001 & 100100 & 110110 & 001001 & 111111 \\
000111 & 000000 & 000000 & 000000 & 000000 & 000000 & 100011 & 111000 & 100011 & 010010 & 111000 & 101010 \\
000000 & 100011 & 000000 & 000000 & 000000 & 000000 & 100100 & 000111 & 001110 & 000111 & 111111 & 101101 \\
000000 & 010010 & 000000 & 000000 & 000000 & 000000 & 110110 & 100100 & 001001 & 100100 & 000000 & 011011 \\
000000 & 001001 & 000000 & 000000 & 000000 & 000000 & 011011 & 010010 & 100100 & 010010 & 000000 & 101101 \\
000000 & 000111 & 000000 & 000000 & 000000 & 000000 & 001001 & 001110 & 011100 & 001110 & 111111 & 011011 \\
000000 & 000000 & 100011 & 000000 & 000000 & 000000 & 010010 & 110110 & 110001 & 101010 & 001110 & 010010 \\
000000 & 000000 & 010010 & 000000 & 000000 & 000000 & 011011 & 101101 & 001001 & 111111 & 001001 & 011011 \\
000000 & 000000 & 001001 & 000000 & 000000 & 000000 & 101101 & 110110 & 100100 & 111111 & 100100 & 101101 \\
000000 & 000000 & 000111 & 000000 & 000000 & 000000 & 100100 & 101101 & 100011 & 010101 & 011100 & 100100 \\
000000 & 000000 & 000000 & 100011 & 000000 & 000000 & 000000 & 001001 & 110001 & 000111 & 010010 & 011100 \\
000000 & 000000 & 000000 & 010010 & 000000 & 000000 & 000000 & 101101 & 001001 & 100100 & 011011 & 010010 \\
000000 & 000000 & 000000 & 001001 & 000000 & 000000 & 000000 & 110110 & 100100 & 010010 & 101101 & 001001 \\
000000 & 000000 & 000000 & 000111 & 000000 & 000000 & 000000 & 010010 & 100011 & 001110 & 100100 & 111000 \\
000000 & 000000 & 000000 & 000000 & 100011 & 000000 & 011100 & 111111 & 010101 & 000111 & 000000 & 001001 \\
000000 & 000000 & 000000 & 000000 & 010010 & 000000 & 010010 & 000000 & 111111 & 100100 & 000000 & 101101 \\
000000 & 000000 & 000000 & 000000 & 001001 & 000000 & 001001 & 000000 & 111111 & 010010 & 000000 & 110110 \\
000000 & 000000 & 000000 & 000000 & 000111 & 000000 & 111000 & 111111 & 101010 & 001110 & 000000 & 010010 \\
000000 & 000000 & 000000 & 000000 & 000000 & 100011 & 000111 & 110001 & 100100 & 100011 & 010101 & 011100 \\
000000 & 000000 & 000000 & 000000 & 000000 & 010010 & 100100 & 001001 & 110110 & 010010 & 111111 & 010010 \\
000000 & 000000 & 000000 & 000000 & 000000 & 001001 & 010010 & 100100 & 011011 & 001001 & 111111 & 001001 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000111 & 001110 & 100011 & 001001 & 000111 & 101010 & 111000
\end{bmatrix}$$

$$G_{13}^{\times} = \begin{bmatrix}
100011 & 000000 & 000000 & 000000 & 000000 & 000000 & 001001 & 111000 & 100011 & 111111 & 110110 & 100011 \\
010010 & 000000 & 000000 & 000000 & 000000 & 000000 & 101101 & 100100 & 010010 & 000000 & 101101 & 010010 \\
001001 & 000000 & 000000 & 000000 & 000000 & 000000 & 110110 & 010010 & 001001 & 000000 & 110110 & 001001 \\
000111 & 000000 & 000000 & 000000 & 000000 & 000000 & 010010 & 110001 & 000111 & 111111 & 101101 & 000111 \\
000000 & 100011 & 000000 & 000000 & 000000 & 000000 & 111000 & 101010 & 111000 & 000111 & 001110 & 100100 \\
000000 & 010010 & 000000 & 000000 & 000000 & 000000 & 100100 & 111111 & 100100 & 100100 & 001001 & 110110 \\
000000 & 001001 & 000000 & 000000 & 000000 & 000000 & 010010 & 111111 & 010010 & 010010 & 100100 & 011011 \\
000000 & 000111 & 000000 & 000000 & 000000 & 000000 & 110001 & 010101 & 110001 & 001110 & 011100 & 001001 \\
000000 & 000000 & 100011 & 000000 & 000000 & 000000 & 010101 & 001110 & 111111 & 110110 & 010010 & 010101 \\
000000 & 000000 & 010010 & 000000 & 000000 & 000000 & 111111 & 001001 & 000000 & 101101 & 011011 & 111111 \\
000000 & 000000 & 001001 & 000000 & 000000 & 000000 & 111111 & 100100 & 000000 & 110110 & 101101 & 111111 \\
000000 & 000000 & 000111 & 000000 & 000000 & 000000 & 101010 & 011100 & 111111 & 101101 & 100100 & 101010 \\
000000 & 000000 & 000000 & 100011 & 000000 & 000000 & 001001 & 111000 & 101101 & 000111 & 010010 & 101010 \\
000000 & 000000 & 000000 & 010010 & 000000 & 000000 & 101101 & 100100 & 011011 & 100100 & 011011 & 111111 \\
000000 & 000000 & 000000 & 001001 & 000000 & 000000 & 110110 & 010010 & 101101 & 010010 & 101101 & 111111 \\
000000 & 000000 & 000000 & 000111 & 000000 & 000000 & 010010 & 110001 & 011011 & 001110 & 100100 & 010101 \\
000000 & 000000 & 000000 & 000000 & 100011 & 000000 & 110110 & 100011 & 001001 & 010010 & 010101 & 110001 \\
000000 & 000000 & 000000 & 000000 & 010010 & 000000 & 101101 & 010010 & 101101 & 011011 & 111111 & 001001 \\
000000 & 000000 & 000000 & 000000 & 001001 & 000000 & 110110 & 001001 & 110110 & 101101 & 111111 & 100100 \\
000000 & 000000 & 000000 & 000000 & 000111 & 000000 & 101101 & 000111 & 010010 & 100100 & 101010 & 100011 \\
000000 & 000000 & 000000 & 000000 & 000000 & 100011 & 100011 & 001001 & 001110 & 100011 & 011100 & 101010 \\
000000 & 000000 & 000000 & 000000 & 000000 & 010010 & 010010 & 101101 & 001001 & 010010 & 010010 & 111111 \\
000000 & 000000 & 000000 & 000000 & 000000 & 001001 & 001001 & 110110 & 100100 & 001001 & 001001 & 111111 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000111 & 000111 & 010010 & 011100 & 000111 & 111000 & 010101
\end{bmatrix}$$

$$G_{14}^{\times} = \begin{bmatrix}
100011 & 000000 & 000000 & 000000 & 000000 & 000000 & 001001 & 111000 & 011100 & 000000 & 001001 & 011100 \\
010010 & 000000 & 000000 & 000000 & 000000 & 000000 & 101101 & 100100 & 010010 & 000000 & 101101 & 010010 \\
001001 & 000000 & 000000 & 000000 & 000000 & 000000 & 110110 & 010010 & 001001 & 000000 & 110110 & 001001 \\
000111 & 000000 & 000000 & 000000 & 000000 & 000000 & 010010 & 110001 & 111000 & 000000 & 010010 & 111000 \\
000000 & 100011 & 000000 & 000000 & 000000 & 000000 & 111000 & 101010 & 000111 & 111000 & 001110 & 100100 \\
000000 & 010010 & 000000 & 000000 & 000000 & 000000 & 100100 & 111111 & 100100 & 100100 & 001001 & 110110 \\
000000 & 001001 & 000000 & 000000 & 000000 & 000000 & 010010 & 111111 & 010010 & 010010 & 100100 & 011011 \\
000000 & 000111 & 000000 & 000000 & 000000 & 000000 & 110001 & 010101 & 001110 & 110001 & 011100 & 001001 \\
000000 & 000000 & 100011 & 000000 & 000000 & 000000 & 010101 & 001110 & 000000 & 001001 & 010010 & 010101 \\
000000 & 000000 & 010010 & 000000 & 000000 & 000000 & 111111 & 001001 & 000000 & 101101 & 011011 & 111111 \\
000000 & 000000 & 001001 & 000000 & 000000 & 000000 & 111111 & 100100 & 000000 & 110110 & 101101 & 111111 \\
000000 & 000000 & 000111 & 000000 & 000000 & 000000 & 101010 & 011100 & 000000 & 010010 & 100100 & 101010 \\
000000 & 000000 & 000000 & 100011 & 000000 & 000000 & 001001 & 111000 & 010010 & 111000 & 101101 & 010101 \\
000000 & 000000 & 000000 & 010010 & 000000 & 000000 & 101101 & 100100 & 011011 & 100100 & 011011 & 111111 \\
000000 & 000000 & 000000 & 001001 & 000000 & 000000 & 110110 & 010010 & 101101 & 010010 & 101101 & 111111 \\
000000 & 000000 & 000000 & 000111 & 000000 & 000000 & 010010 & 110001 & 100100 & 110001 & 011011 & 101010 \\
000000 & 000000 & 000000 & 000000 & 100011 & 000000 & 110110 & 100011 & 001001 & 010010 & 010101 & 110001 \\
000000 & 000000 & 000000 & 000000 & 010010 & 000000 & 101101 & 010010 & 101101 & 011011 & 111111 & 001001 \\
000000 & 000000 & 000000 & 000000 & 001001 & 000000 & 110110 & 001001 & 110110 & 101101 & 111111 & 100100 \\
000000 & 000000 & 000000 & 000000 & 000111 & 000000 & 101101 & 000111 & 010010 & 100100 & 101010 & 100011 \\
000000 & 000000 & 000000 & 000000 & 000000 & 100011 & 100011 & 001001 & 001110 & 100011 & 011100 & 101010 \\
000000 & 000000 & 000000 & 000000 & 000000 & 010010 & 010010 & 101101 & 001001 & 010010 & 010010 & 111111 \\
000000 & 000000 & 000000 & 000000 & 000000 & 001001 & 001001 & 110110 & 100100 & 001001 & 001001 & 111111 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000111 & 000111 & 010010 & 011100 & 000111 & 111000 & 010101
\end{bmatrix}$$

$$G_{15}^{\times} = \begin{bmatrix}
100011 & 000000 & 000000 & 000000 & 000000 & 000000 & 110110 & 111000 & 011100 & 111111 & 001001 & 011100 \\
010010 & 000000 & 000000 & 000000 & 000000 & 000000 & 101101 & 100100 & 010010 & 000000 & 101101 & 010010 \\
001001 & 000000 & 000000 & 000000 & 000000 & 000000 & 110110 & 010010 & 001001 & 000000 & 110110 & 001001 \\
000111 & 000000 & 000000 & 000000 & 000000 & 000000 & 101101 & 110001 & 111000 & 111111 & 010010 & 111000 \\
000000 & 100011 & 000000 & 000000 & 000000 & 000000 & 111000 & 101010 & 000111 & 000111 & 110001 & 100100 \\
000000 & 010010 & 000000 & 000000 & 000000 & 000000 & 100100 & 111111 & 100100 & 100100 & 001001 & 110110 \\
000000 & 001001 & 000000 & 000000 & 000000 & 000000 & 010010 & 111111 & 010010 & 010010 & 100100 & 011011 \\
000000 & 000111 & 000000 & 000000 & 000000 & 000000 & 110001 & 010101 & 001110 & 001110 & 100011 & 001001 \\
000000 & 000000 & 100011 & 000000 & 000000 & 000000 & 101010 & 110001 & 111111 & 110110 & 010010 & 101010 \\
000000 & 000000 & 010010 & 000000 & 000000 & 000000 & 111111 & 001001 & 000000 & 101101 & 011011 & 111111 \\
000000 & 000000 & 001001 & 000000 & 000000 & 000000 & 111111 & 100100 & 000000 & 110110 & 101101 & 111111 \\
000000 & 000000 & 000111 & 000000 & 000000 & 000000 & 010101 & 100011 & 111111 & 101101 & 100100 & 010101 \\
000000 & 000000 & 000000 & 100011 & 000000 & 000000 & 001001 & 111000 & 010010 & 000111 & 101101 & 101010 \\
000000 & 000000 & 000000 & 010010 & 000000 & 000000 & 101101 & 100100 & 011011 & 100100 & 011011 & 111111 \\
000000 & 000000 & 000000 & 001001 & 000000 & 000000 & 110110 & 010010 & 101101 & 010010 & 101101 & 111111 \\
000000 & 000000 & 000000 & 000111 & 000000 & 000000 & 010010 & 110001 & 100100 & 001110 & 011011 & 010101 \\
000000 & 000000 & 000000 & 000000 & 100011 & 000000 & 001001 & 100011 & 110110 & 010010 & 101010 & 001110 \\
000000 & 000000 & 000000 & 000000 & 010010 & 000000 & 101101 & 010010 & 101101 & 011011 & 111111 & 001001 \\
000000 & 000000 & 000000 & 000000 & 001001 & 000000 & 110110 & 001001 & 110110 & 101101 & 111111 & 100100 \\
000000 & 000000 & 000000 & 000000 & 000111 & 000000 & 010010 & 000111 & 101101 & 100100 & 010101 & 011100 \\
000000 & 000000 & 000000 & 000000 & 000000 & 100011 & 100011 & 001001 & 001110 & 100011 & 011100 & 101010 \\
000000 & 000000 & 000000 & 000000 & 000000 & 010010 & 010010 & 101101 & 001001 & 010010 & 010010 & 111111 \\
000000 & 000000 & 000000 & 000000 & 000000 & 001001 & 001001 & 110110 & 100100 & 001001 & 001001 & 111111 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000111 & 000111 & 010010 & 011100 & 000111 & 111000 & 010101
\end{bmatrix}$$

$$G_{16}^{\times} = \begin{bmatrix}
100011 & 000000 & 000000 & 000000 & 000000 & 000111 & 010010 & 110001 & 011011 & 011100 & 111000 & 110001 \\
010010 & 000000 & 000000 & 000000 & 000000 & 000000 & 011100 & 100011 & 011100 & 010010 & 110110 & 101010 \\
001001 & 000000 & 000000 & 000000 & 000000 & 000000 & 001110 & 110001 & 001110 & 001001 & 011011 & 010101 \\
000111 & 000000 & 000000 & 000000 & 000000 & 000111 & 010101 & 001001 & 011100 & 111000 & 010101 & 011011 \\
000000 & 100011 & 000000 & 000000 & 000000 & 000000 & 000111 & 000111 & 111111 & 011011 & 001110 & 110001 & 101010 \\
000000 & 010010 & 000000 & 000000 & 000000 & 000000 & 011100 & 010101 & 100011 & 001001 & 011011 & 100011 \\
000000 & 001001 & 000000 & 000000 & 000000 & 000000 & 001110 & 101010 & 110001 & 100100 & 101101 & 110001 \\
000000 & 000111 & 000000 & 000000 & 000000 & 000111 & 000000 & 101010 & 100011 & 011100 & 000111 & 010010 \\
000000 & 000000 & 100011 & 000000 & 000000 & 000000 & 001001 & 000000 & 110001 & 010010 & 001110 & 100011 \\
000000 & 000000 & 010010 & 000000 & 000000 & 000000 & 010010 & 111111 & 110110 & 011011 & 001001 & 101101 \\
000000 & 000000 & 001001 & 000000 & 000000 & 000000 & 001001 & 111111 & 011011 & 101101 & 100100 & 110110 \\
000000 & 000000 & 000111 & 000000 & 000000 & 000000 & 010010 & 000000 & 100011 & 100100 & 011100 & 000111 \\
000000 & 000000 & 000000 & 100011 & 000000 & 000000 & 010101 & 110001 & 110110 & 111000 & 100100 & 010010 \\
000000 & 000000 & 000000 & 010010 & 000000 & 000000 & 000000 & 110110 & 010010 & 100100 & 110110 & 100100 \\
000000 & 000000 & 000000 & 001001 & 000000 & 000000 & 000000 & 011011 & 001001 & 010010 & 011011 & 010010 \\
000000 & 000000 & 000000 & 000111 & 000000 & 000000 & 010101 & 011100 & 010010 & 110001 & 001001 & 011011 \\
000000 & 000000 & 000000 & 000000 & 100011 & 000111 & 000111 & 110001 & 010010 & 100100 & 110110 & 110110 \\
000000 & 000000 & 000000 & 000000 & 010010 & 000000 & 011100 & 011100 & 001110 & 110110 & 111111 & 110001 \\
000000 & 000000 & 000000 & 000000 & 001001 & 000000 & 001110 & 001110 & 000111 & 011011 & 111111 & 111000 \\
000000 & 000000 & 000000 & 000000 & 000111 & 000111 & 000000 & 110110 & 110001 & 001001 & 001001 & 101010 \\
000000 & 000000 & 000000 & 000000 & 000000 & 100100 & 000111 & 101010 & 101010 & 000000 & 010010 & 100011 \\
000000 & 000000 & 000000 & 000000 & 000000 & 010010 & 011100 & 101010 & 101010 & 000000 & 001001 & 001110 \\
000000 & 000000 & 000000 & 000000 & 000000 & 001001 & 001110 & 010101 & 010101 & 000000 & 100100 & 000111 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 111111 & 111111 & 111111 & 000000 & 000000 & 111111
\end{bmatrix}$$

$$G_{17}^{\times} = \begin{bmatrix}
100011 & 000000 & 000000 & 000000 & 000000 & 000111 & 010010 & 110001 & 011011 & 100011 & 000111 & 110001 \\
010010 & 000000 & 000000 & 000000 & 000000 & 000000 & 011100 & 100011 & 011100 & 101101 & 001001 & 101010 \\
001001 & 000000 & 000000 & 000000 & 000000 & 000000 & 001110 & 110001 & 001110 & 110110 & 100100 & 010101 \\
000111 & 000000 & 000000 & 000000 & 000000 & 000111 & 010101 & 001001 & 011100 & 111000 & 010101 & 011011 \\
000000 & 100011 & 000000 & 000000 & 000000 & 000111 & 000111 & 111111 & 011011 & 110001 & 001110 & 101010 \\
000000 & 010010 & 000000 & 000000 & 000000 & 000000 & 011100 & 010101 & 100011 & 110110 & 100100 & 100011 \\
000000 & 001001 & 000000 & 000000 & 000000 & 000000 & 001110 & 101010 & 110001 & 011011 & 010010 & 110001 \\
000000 & 000111 & 000000 & 000000 & 000000 & 000111 & 000000 & 101010 & 100011 & 011100 & 000111 & 010010 \\
000000 & 000000 & 100011 & 000000 & 000000 & 000000 & 011100 & 010101 & 100100 & 010010 & 001110 & 110110 \\
000000 & 000000 & 010010 & 000000 & 000000 & 000000 & 010010 & 111111 & 110110 & 011011 & 001001 & 101101 \\
000000 & 000000 & 001001 & 000000 & 000000 & 000000 & 001001 & 111111 & 011011 & 101101 & 100100 & 110110 \\
000000 & 000000 & 000111 & 000000 & 000000 & 000000 & 000111 & 010101 & 110110 & 100100 & 011100 & 010010 \\
000000 & 000000 & 000000 & 100011 & 000000 & 000000 & 000000 & 100100 & 100011 & 111000 & 100100 & 000111 \\
000000 & 000000 & 000000 & 010010 & 000000 & 000000 & 000000 & 110110 & 010010 & 100100 & 110110 & 100100 \\
000000 & 000000 & 000000 & 001001 & 000000 & 000000 & 000000 & 011011 & 001001 & 010010 & 011011 & 010010 \\
000000 & 000000 & 000000 & 000111 & 000000 & 000000 & 000000 & 001001 & 000111 & 110001 & 001001 & 001110 \\
000000 & 000000 & 000000 & 000000 & 100011 & 000111 & 000111 & 110001 & 010010 & 011011 & 001001 & 110110 \\
000000 & 000000 & 000000 & 000000 & 010010 & 000000 & 011100 & 011100 & 001110 & 001001 & 000000 & 110001 \\
000000 & 000000 & 000000 & 000000 & 001001 & 000000 & 001110 & 001110 & 000111 & 100100 & 000000 & 111000 \\
000000 & 000000 & 000000 & 000000 & 000111 & 000111 & 000000 & 110110 & 110001 & 001001 & 001001 & 101010 \\
000000 & 000000 & 000000 & 000000 & 000000 & 100100 & 000111 & 101010 & 101010 & 111111 & 101101 & 100011 \\
000000 & 000000 & 000000 & 000000 & 000000 & 010010 & 011100 & 101010 & 101010 & 111111 & 110110 & 001110 \\
000000 & 000000 & 000000 & 000000 & 000000 & 001001 & 001110 & 010101 & 010101 & 111111 & 011011 & 000111 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 111111 & 111111 & 111111 & 000000 & 000000 & 111111
\end{bmatrix}$$

$$G_{18}^{\times} = \begin{bmatrix}
100011 & 000000 & 000000 & 000000 & 000000 & 001110 & 001001 & 001110 & 011011 & 011100 & 100011 & 011100 \\
010010 & 000000 & 000000 & 000000 & 000000 & 001001 & 000111 & 100011 & 100011 & 101101 & 101101 & 111000 \\
001001 & 000000 & 000000 & 000000 & 000000 & 001001 & 000000 & 100100 & 011011 & 001001 & 111111 & 010010 \\
000111 & 000000 & 000000 & 000000 & 000000 & 000111 & 000000 & 100011 & 001001 & 111000 & 101010 & 110001 \\
000000 & 100011 & 000000 & 000000 & 000000 & 001110 & 001001 & 101010 & 110001 & 001110 & 010101 & 101101 \\
000000 & 010010 & 000000 & 000000 & 000000 & 001001 & 000111 & 010101 & 011100 & 110110 & 000000 & 110001 \\
000000 & 001001 & 000000 & 000000 & 000000 & 001001 & 000000 & 111111 & 100100 & 100100 & 001001 & 110110 \\
000000 & 000111 & 000000 & 000000 & 000000 & 000111 & 000000 & 101010 & 011100 & 011100 & 000111 & 010010 \\
000000 & 000000 & 100011 & 000000 & 000000 & 001001 & 000111 & 101010 & 100100 & 101101 & 101010 & 011011 \\
000000 & 000000 & 010010 & 000000 & 000000 & 000000 & 000111 & 101010 & 011100 & 100100 & 110110 & 111000 \\
000000 & 000000 & 001001 & 000000 & 000000 & 000000 & 001001 & 111111 & 011011 & 101101 & 100100 & 110110 \\
000000 & 000000 & 000111 & 000000 & 000000 & 001001 & 001001 & 111111 & 011100 & 100100 & 000111 & 101010 \\
000000 & 000000 & 000000 & 100011 & 000000 & 001001 & 001110 & 110001 & 110110 & 000111 & 111111 & 000000 \\
000000 & 000000 & 000000 & 010010 & 000000 & 000000 & 000000 & 110110 & 010010 & 100100 & 110110 & 100100 \\
000000 & 000000 & 000000 & 001001 & 000000 & 000000 & 000000 & 011011 & 001001 & 010010 & 011011 & 010010 \\
000000 & 000000 & 000000 & 000111 & 000000 & 001001 & 001110 & 011100 & 010010 & 001110 & 010010 & 001001 \\
000000 & 000000 & 000000 & 000000 & 100011 & 001110 & 001001 & 011011 & 111000 & 011011 & 010010 & 001110 \\
000000 & 000000 & 000000 & 000000 & 010010 & 001001 & 000111 & 011100 & 110001 & 001001 & 100100 & 100011 \\
000000 & 000000 & 000000 & 000000 & 001001 & 001001 & 000000 & 011011 & 010010 & 011011 & 011011 & 111111 \\
000000 & 000000 & 000000 & 000000 & 000111 & 000111 & 000000 & 001001 & 001110 & 110110 & 001001 & 010101 \\
000000 & 000000 & 000000 & 000000 & 000000 & 101101 & 001001 & 111111 & 111111 & 000000 & 110110 & 100100 \\
000000 & 000000 & 000000 & 000000 & 000000 & 011011 & 000111 & 101010 & 010101 & 111111 & 010010 & 011100 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 101010 & 101010 & 010101 & 111111 & 111111 & 101010 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 010101 & 010101 & 101010 & 111111 & 111111 & 010101
\end{bmatrix}$$

$$G_{19}^{\times} = \begin{bmatrix}
100011 & 000000 & 000000 & 000000 & 000000 & 010010 & 110001 & 001110 & 001001 & 110110 & 000111 & 110110 \\
010010 & 000000 & 000000 & 000000 & 000000 & 011011 & 001001 & 001001 & 000000 & 000000 & 010010 & 110110 \\
001001 & 000000 & 000000 & 000000 & 000000 & 010010 & 011011 & 001110 & 000000 & 101010 & 100011 & 001110 \\
000111 & 000000 & 000000 & 000000 & 000000 & 011011 & 011100 & 001001 & 001001 & 100011 & 110110 & 110001 \\
000000 & 100011 & 000000 & 000000 & 000000 & 001001 & 101101 & 001001 & 000111 & 011100 & 110110 & 000111 \\
000000 & 010010 & 000000 & 000000 & 000000 & 010010 & 100100 & 000111 & 001001 & 010101 & 110001 & 101010 \\
000000 & 001001 & 000000 & 000000 & 000000 & 001001 & 010010 & 011100 & 001001 & 111000 & 110001 & 110001 \\
000000 & 000111 & 000000 & 000000 & 000000 & 010010 & 011011 & 010010 & 001110 & 111000 & 101101 & 001110 \\
000000 & 000000 & 100011 & 000000 & 000000 & 001110 & 101101 & 000111 & 001001 & 010010 & 110001 & 011011 \\
000000 & 000000 & 010010 & 000000 & 000000 & 001001 & 011011 & 011011 & 000000 & 001001 & 000000 & 010010 \\
000000 & 000000 & 001001 & 000000 & 000000 & 011011 & 010010 & 000111 & 000000 & 001110 & 101010 & 011100 \\
000000 & 000000 & 000111 & 000000 & 000000 & 011100 & 011011 & 001110 & 001001 & 111111 & 001110 & 000000 \\
000000 & 000000 & 000000 & 100011 & 000000 & 000111 & 111000 & 011011 & 000111 & 110001 & 100011 & 001001 \\
000000 & 000000 & 000000 & 010010 & 000000 & 011011 & 011011 & 011100 & 001001 & 001110 & 001110 & 100011 \\
000000 & 000000 & 000000 & 001001 & 000000 & 010010 & 010010 & 011011 & 001001 & 111111 & 100100 & 000000 \\
000000 & 000000 & 000000 & 000111 & 000000 & 001110 & 110001 & 001001 & 001110 & 011100 & 111000 & 101101 \\
000000 & 000000 & 000000 & 000000 & 100011 & 011100 & 000111 & 000111 & 001110 & 111000 & 001001 & 110110 \\
000000 & 000000 & 000000 & 000000 & 010010 & 010010 & 100100 & 011011 & 001001 & 011011 & 010010 & 010010 \\
000000 & 000000 & 000000 & 000000 & 001001 & 001001 & 010010 & 010010 & 001001 & 111111 & 000000 & 101101 \\
000000 & 000000 & 000000 & 000000 & 000111 & 000111 & 110001 & 011011 & 000111 & 111111 & 101010 & 110001 \\
000000 & 000000 & 000000 & 000000 & 000000 & 111111 & 111111 & 010101 & 000000 & 010101 & 010101 & 101010 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 111111 & 000000 & 111111 & 111111 & 111111 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 101101 & 101101 & 110110 & 011011 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 011011 & 011011 & 101101 & 110110
\end{bmatrix}$$

$$G_{20}^{\times} = \begin{bmatrix}
100011 & 000000 & 000000 & 000000 & 000000 & 010010 & 110001 & 011011 & 001001 & 100011 & 101101 & 100011 \\
010010 & 000000 & 000000 & 000000 & 000000 & 011011 & 001001 & 001001 & 000000 & 000000 & 010010 & 110110 \\
001001 & 000000 & 000000 & 000000 & 000000 & 010010 & 011011 & 001110 & 000000 & 101010 & 100011 & 001110 \\
000111 & 000000 & 000000 & 000000 & 000000 & 011011 & 011100 & 011100 & 001001 & 110110 & 011100 & 100100 \\
000000 & 100011 & 000000 & 000000 & 000000 & 001001 & 101101 & 011100 & 001110 & 000000 & 111000 & 111111 \\
000000 & 010010 & 000000 & 000000 & 000000 & 010010 & 100100 & 000111 & 001001 & 010101 & 110001 & 101010 \\
000000 & 001001 & 000000 & 000000 & 000000 & 001001 & 010010 & 011100 & 001001 & 111000 & 110001 & 110001 \\
000000 & 000111 & 000000 & 000000 & 000000 & 010010 & 011011 & 000111 & 000111 & 100100 & 100011 & 110110 \\
000000 & 000000 & 100011 & 000000 & 000000 & 001110 & 101101 & 010010 & 001001 & 000111 & 011011 & 001110 \\
000000 & 000000 & 010010 & 000000 & 000000 & 001001 & 011011 & 011011 & 000000 & 001001 & 000000 & 010010 \\
000000 & 000000 & 001001 & 000000 & 000000 & 011011 & 010010 & 000111 & 000000 & 001110 & 101010 & 011100 \\
000000 & 000000 & 000111 & 000000 & 000000 & 011100 & 011011 & 011011 & 001001 & 101010 & 100100 & 010101 \\
000000 & 000000 & 000000 & 100011 & 000000 & 000111 & 111000 & 011011 & 001110 & 111000 & 000111 & 100100 \\
000000 & 000000 & 000000 & 010010 & 000000 & 011011 & 011011 & 011100 & 001001 & 001110 & 001110 & 100011 \\
000000 & 000000 & 000000 & 001001 & 000000 & 010010 & 010010 & 011011 & 001001 & 111111 & 100100 & 000000 \\
000000 & 000000 & 000000 & 000111 & 000000 & 001110 & 110001 & 001001 & 000111 & 010101 & 011100 & 000000 \\
000000 & 000000 & 000000 & 000000 & 100011 & 011100 & 000111 & 000111 & 001110 & 111000 & 001001 & 110110 \\
000000 & 000000 & 000000 & 000000 & 010010 & 010010 & 100100 & 011011 & 001001 & 011011 & 010010 & 010010 \\
000000 & 000000 & 000000 & 000000 & 001001 & 001001 & 010010 & 010010 & 001001 & 111111 & 000000 & 101101 \\
000000 & 000000 & 000000 & 000000 & 000111 & 000111 & 110001 & 011011 & 000111 & 111111 & 101010 & 110001 \\
000000 & 000000 & 000000 & 000000 & 000000 & 111111 & 111111 & 010101 & 000000 & 010101 & 010101 & 101010 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 111111 & 000000 & 111111 & 111111 & 111111 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 101101 & 101101 & 110110 & 011011 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 011011 & 011011 & 101101 & 110110
\end{bmatrix}$$

$$G_{21}^{\times} = \begin{bmatrix}
100011 & 000000 & 000000 & 000000 & 000000 & 001001 & 001110 & 110001 & 011100 & 100100 & 100100 & 010010 \\
010010 & 000000 & 000000 & 000000 & 000000 & 000000 & 000111 & 011100 & 000000 & 110001 & 100100 & 011100 \\
001001 & 000000 & 000000 & 000000 & 000000 & 000000 & 001001 & 100100 & 111111 & 101101 & 101101 & 011011 \\
000111 & 000000 & 000000 & 000000 & 000000 & 001001 & 000000 & 001001 & 011100 & 000111 & 101101 & 101010 \\
000000 & 100011 & 000000 & 000000 & 000000 & 000111 & 001001 & 011100 & 000111 & 110110 & 001110 & 011100 \\
000000 & 010010 & 000000 & 000000 & 000000 & 001001 & 001001 & 101101 & 001001 & 111111 & 100100 & 000000 \\
000000 & 001001 & 000000 & 000000 & 000000 & 001001 & 000000 & 001001 & 001001 & 101101 & 111111 & 010010 \\
000000 & 000111 & 000000 & 000000 & 000000 & 001110 & 000111 & 101101 & 110001 & 000111 & 100011 & 010010 \\
000000 & 000000 & 100011 & 000000 & 000000 & 001001 & 000111 & 110001 & 011011 & 100100 & 100011 & 110110 \\
000000 & 000000 & 010010 & 000000 & 000000 & 000000 & 000000 & 110110 & 011011 & 100100 & 111111 & 111111 \\
000000 & 000000 & 001001 & 000000 & 000000 & 000000 & 000000 & 011011 & 101101 & 010010 & 111111 & 111111 \\
000000 & 000000 & 000111 & 000000 & 000000 & 001001 & 000111 & 011100 & 101101 & 101101 & 011100 & 001001 \\
000000 & 000000 & 000000 & 100011 & 000000 & 000111 & 001001 & 101101 & 101101 & 110001 & 111111 & 010101 \\
000000 & 000000 & 000000 & 010010 & 000000 & 001001 & 001001 & 100100 & 110110 & 011011 & 101101 & 101101 \\
000000 & 000000 & 000000 & 001001 & 000000 & 001001 & 000000 & 101101 & 110110 & 111111 & 011011 & 100100 \\
000000 & 000000 & 000000 & 000111 & 000000 & 001110 & 000111 & 001110 & 100100 & 001001 & 000000 & 000000 \\
000000 & 000000 & 000000 & 000000 & 100011 & 001110 & 001110 & 101101 & 111000 & 100100 & 111000 & 111000 \\
000000 & 000000 & 000000 & 000000 & 010010 & 001001 & 001001 & 011011 & 100100 & 110110 & 100100 & 100100 \\
000000 & 000000 & 000000 & 000000 & 001001 & 001001 & 000000 & 010010 & 111111 & 001001 & 111111 & 000000 \\
000000 & 000000 & 000000 & 000000 & 000111 & 000111 & 000000 & 110001 & 010101 & 000111 & 010101 & 111111 \\
000000 & 000000 & 000000 & 000000 & 000000 & 101101 & 001110 & 010101 & 010010 & 000111 & 010010 & 000111 \\
000000 & 000000 & 000000 & 000000 & 000000 & 011011 & 001001 & 111111 & 011011 & 100100 & 011011 & 100100 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 101010 & 101010 & 111111 & 010101 & 111111 & 010101 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 010101 & 010101 & 111111 & 101010 & 111111 & 101010
\end{bmatrix}$$

$$G_{22}^{\times} = \begin{bmatrix}
100011 & 000000 & 000000 & 000000 & 000000 & 000000 & 011100 & 110001 & 010101 & 010010 & 010010 & 011011 \\
010010 & 000000 & 000000 & 000000 & 000000 & 000000 & 010010 & 001001 & 111111 & 011011 & 011011 & 110110 \\
001001 & 000000 & 000000 & 000000 & 000000 & 000000 & 001001 & 100100 & 111111 & 101101 & 101101 & 011011 \\
000111 & 000000 & 000000 & 000000 & 000000 & 000000 & 000111 & 011100 & 101010 & 011011 & 100100 & 001001 \\
000000 & 100011 & 000000 & 000000 & 000000 & 000111 & 011100 & 001001 & 000111 & 100011 & 001110 & 110110 \\
000000 & 010010 & 000000 & 000000 & 000000 & 000000 & 001110 & 111000 & 111111 & 100011 & 101101 & 100011 \\
000000 & 001001 & 000000 & 000000 & 000000 & 000000 & 000111 & 011100 & 111111 & 110001 & 110110 & 110001 \\
000000 & 000111 & 000000 & 000000 & 000000 & 000111 & 000000 & 111000 & 111000 & 100100 & 010101 & 110001 \\
000000 & 000000 & 100011 & 000000 & 000000 & 000000 & 000000 & 100100 & 010010 & 000111 & 010101 & 010101 \\
000000 & 000000 & 010010 & 000000 & 000000 & 000000 & 000000 & 110110 & 011011 & 100100 & 111111 & 111111 \\
000000 & 000000 & 001001 & 000000 & 000000 & 000000 & 000000 & 011011 & 101101 & 010010 & 111111 & 111111 \\
000000 & 000000 & 000111 & 000000 & 000000 & 000000 & 000000 & 001001 & 100100 & 001110 & 101010 & 101010 \\
000000 & 000000 & 000000 & 100011 & 000000 & 000111 & 011100 & 111000 & 010010 & 011011 & 000000 & 111111 \\
000000 & 000000 & 000000 & 010010 & 000000 & 000000 & 001110 & 110001 & 000000 & 000111 & 100100 & 001110 \\
000000 & 000000 & 000000 & 001001 & 000000 & 000000 & 000111 & 111000 & 000000 & 100011 & 010010 & 000111 \\
000000 & 000000 & 000000 & 000111 & 000000 & 000111 & 000000 & 011011 & 010010 & 010101 & 001001 & 100011 \\
000000 & 000000 & 000000 & 000000 & 100011 & 000111 & 001001 & 111000 & 001110 & 111000 & 110001 & 011011 \\
000000 & 000000 & 000000 & 000000 & 010010 & 000000 & 001110 & 001110 & 010010 & 101010 & 101101 & 000111 \\
000000 & 000000 & 000000 & 000000 & 001001 & 000000 & 000111 & 000111 & 001001 & 010101 & 110110 & 100011 \\
000000 & 000000 & 000000 & 000000 & 000111 & 000111 & 010101 & 100100 & 101010 & 101101 & 101010 & 010101 \\
000000 & 000000 & 000000 & 000000 & 000000 & 100100 & 011100 & 010101 & 011011 & 110001 & 100100 & 001110 \\
000000 & 000000 & 000000 & 000000 & 000000 & 010010 & 001110 & 101010 & 101101 & 111000 & 010010 & 000111 \\
000000 & 000000 & 000000 & 000000 & 000000 & 001001 & 000111 & 010101 & 110110 & 011100 & 001001 & 100011 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 111111 & 111111 & 000000 & 111111 & 000000 & 111111
\end{bmatrix}$$

$$G_{23}^{\times} = \begin{bmatrix}
100011 & 000000 & 000000 & 000000 & 000000 & 000000 & 000111 & 111111 & 101101 & 100100 & 100011 & 000111 \\
010010 & 000000 & 000000 & 000000 & 000000 & 000000 & 011011 & 000000 & 100100 & 001001 & 010010 & 011011 \\
001001 & 000000 & 000000 & 000000 & 000000 & 000000 & 010010 & 000000 & 101101 & 011011 & 001001 & 010010 \\
000111 & 000000 & 000000 & 000000 & 000000 & 000000 & 001110 & 111111 & 011011 & 001001 & 000111 & 001110 \\
000000 & 100011 & 000000 & 000000 & 000000 & 000111 & 010010 & 001110 & 101101 & 110001 & 000111 & 011100 \\
000000 & 010010 & 000000 & 000000 & 000000 & 000000 & 010101 & 011011 & 101010 & 011100 & 010010 & 010101 \\
000000 & 001001 & 000000 & 000000 & 000000 & 000000 & 010101 & 101101 & 101010 & 110001 & 001001 & 010101 \\
000000 & 000111 & 000000 & 000000 & 000000 & 000111 & 000111 & 111000 & 000111 & 110110 & 100011 & 001001 \\
000000 & 000000 & 100011 & 000000 & 000000 & 000111 & 001001 & 100100 & 111000 & 100011 & 111111 & 111111 \\
000000 & 000000 & 010010 & 000000 & 000000 & 000000 & 011100 & 100100 & 101010 & 111000 & 110110 & 111000 \\
000000 & 000000 & 001001 & 000000 & 000000 & 000000 & 001110 & 010010 & 010101 & 011100 & 011011 & 011100 \\
000000 & 000000 & 000111 & 000000 & 000000 & 000111 & 001110 & 101101 & 010010 & 101101 & 010010 & 110001 \\
000000 & 000000 & 000000 & 100011 & 000000 & 000000 & 000000 & 100011 & 111000 & 110001 & 010010 & 100100 \\
000000 & 000000 & 000000 & 010010 & 000000 & 000000 & 000000 & 010010 & 100100 & 001001 & 011011 & 110110 \\
000000 & 000000 & 000000 & 001001 & 000000 & 000000 & 000000 & 001001 & 010010 & 100100 & 101101 & 011011 \\
000000 & 000000 & 000000 & 000111 & 000000 & 000000 & 000000 & 000111 & 110001 & 100011 & 100100 & 001001 \\
000000 & 000000 & 000000 & 000000 & 100011 & 000111 & 000111 & 000111 & 010010 & 100011 & 001110 & 101101 \\
000000 & 000000 & 000000 & 000000 & 010010 & 000000 & 010101 & 110110 & 010101 & 111000 & 111111 & 100011 \\
000000 & 000000 & 000000 & 000000 & 001001 & 000000 & 010101 & 011011 & 010101 & 100011 & 111111 & 001110 \\
000000 & 000000 & 000000 & 000000 & 000111 & 000111 & 010010 & 101010 & 000111 & 101101 & 110001 & 010101 \\
000000 & 000000 & 000000 & 000000 & 000000 & 100100 & 001110 & 010010 & 110001 & 010101 & 110110 & 000111 \\
000000 & 000000 & 000000 & 000000 & 000000 & 010010 & 000111 & 001001 & 111000 & 101010 & 011011 & 100011 \\
000000 & 000000 & 000000 & 000000 & 000000 & 001001 & 011100 & 100100 & 100011 & 101010 & 101101 & 001110 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 111111 & 000000 & 111111 & 111111 & 000000 & 111111
\end{bmatrix}$$

$$G_{24}^{\times} = \begin{bmatrix}
100011 & 000000 & 000000 & 000000 & 000000 & 000000 & 010010 & 111111 & 111000 & 001110 & 011100 & 101101 \\
010010 & 000000 & 000000 & 000000 & 000000 & 000000 & 011011 & 000000 & 100100 & 001001 & 010010 & 011011 \\
001001 & 000000 & 000000 & 000000 & 000000 & 000000 & 010010 & 000000 & 101101 & 011011 & 001001 & 010010 \\
000111 & 000000 & 000000 & 000000 & 000000 & 000000 & 011011 & 111111 & 001110 & 100011 & 111000 & 100100 \\
000000 & 100011 & 000000 & 000000 & 000000 & 000111 & 010010 & 001110 & 101101 & 001110 & 111000 & 100011 \\
000000 & 010010 & 000000 & 000000 & 000000 & 000000 & 010101 & 011011 & 101010 & 011100 & 010010 & 010101 \\
000000 & 001001 & 000000 & 000000 & 000000 & 000000 & 010101 & 101101 & 101010 & 110001 & 001001 & 010101 \\
000000 & 000111 & 000000 & 000000 & 000000 & 000111 & 000111 & 111000 & 000111 & 001001 & 011100 & 110110 \\
000000 & 000000 & 100011 & 000000 & 000000 & 000111 & 011100 & 100100 & 101101 & 110110 & 111111 & 101010 \\
000000 & 000000 & 010010 & 000000 & 000000 & 000000 & 011100 & 100100 & 101010 & 111000 & 110110 & 111000 \\
000000 & 000000 & 001001 & 000000 & 000000 & 000000 & 001110 & 010010 & 010101 & 011100 & 011011 & 011100 \\
000000 & 000000 & 000111 & 000000 & 000000 & 000111 & 011011 & 101101 & 000111 & 111000 & 010010 & 100100 \\
000000 & 000000 & 000000 & 100011 & 000000 & 000000 & 010101 & 100011 & 101101 & 100100 & 010010 & 110001 \\
000000 & 000000 & 000000 & 010010 & 000000 & 000000 & 000000 & 010010 & 100100 & 001001 & 011011 & 110110 \\
000000 & 000000 & 000000 & 001001 & 000000 & 000000 & 000000 & 001001 & 010010 & 100100 & 101101 & 011011 \\
000000 & 000000 & 000000 & 000111 & 000000 & 000000 & 010101 & 000111 & 100100 & 110110 & 100100 & 011100 \\
000000 & 000000 & 000000 & 000000 & 100011 & 000111 & 000111 & 000111 & 010010 & 100011 & 001110 & 101101 \\
000000 & 000000 & 000000 & 000000 & 010010 & 000000 & 010101 & 110110 & 010101 & 111000 & 111111 & 100011 \\
000000 & 000000 & 000000 & 000000 & 001001 & 000000 & 010101 & 011011 & 010101 & 100011 & 111111 & 001110 \\
000000 & 000000 & 000000 & 000000 & 000111 & 000111 & 010010 & 101010 & 000111 & 101101 & 110001 & 010101 \\
000000 & 000000 & 000000 & 000000 & 000000 & 100100 & 001110 & 010010 & 110001 & 010101 & 110110 & 000111 \\
000000 & 000000 & 000000 & 000000 & 000000 & 010010 & 000111 & 001001 & 111000 & 101010 & 011011 & 100011 \\
000000 & 000000 & 000000 & 000000 & 000000 & 001001 & 011100 & 100100 & 100011 & 101010 & 101101 & 001110 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 111111 & 000000 & 111111 & 111111 & 000000 & 111111
\end{bmatrix}$$

$$G_{25}^{\times} = \begin{bmatrix}
100011 & 000000 & 000000 & 000000 & 001001 & 000000 & 000000 & 000000 & 001001 & 010101 & 011100 & 010101 \\
010010 & 000000 & 000000 & 000000 & 000111 & 000111 & 000000 & 000111 & 111000 & 001110 & 101101 & 110001 \\
001001 & 000000 & 000000 & 000000 & 001001 & 001001 & 000000 & 001001 & 001001 & 101101 & 001001 & 101101 \\
000111 & 000000 & 000000 & 000000 & 000111 & 001110 & 000000 & 001110 & 111000 & 001001 & 000111 & 110110 \\
000000 & 100011 & 000000 & 000000 & 001001 & 000000 & 000000 & 111111 & 111000 & 010101 & 001110 & 000000 \\
000000 & 010010 & 000000 & 000000 & 000111 & 000111 & 000000 & 000111 & 110001 & 001110 & 110110 & 001110 \\
000000 & 001001 & 000000 & 000000 & 001001 & 001001 & 000000 & 001001 & 101101 & 101101 & 100100 & 010010 \\
000000 & 000111 & 000000 & 000000 & 000111 & 001110 & 000000 & 110001 & 011011 & 001001 & 100011 & 011100 \\
000000 & 000000 & 100011 & 000000 & 000111 & 000000 & 000000 & 010101 & 011100 & 111111 & 001110 & 010101 \\
000000 & 000000 & 010010 & 000000 & 001110 & 000111 & 000000 & 111000 & 000111 & 110001 & 110110 & 110001 \\
000000 & 000000 & 001001 & 000000 & 000111 & 001110 & 000000 & 110001 & 100011 & 100011 & 011011 & 100011 \\
000000 & 000000 & 000111 & 000000 & 001110 & 000000 & 000000 & 101010 & 111000 & 111111 & 011100 & 101010 \\
000000 & 000000 & 000000 & 100011 & 000000 & 000111 & 000000 & 010101 & 010101 & 001001 & 010101 & 000111 \\
000000 & 000000 & 000000 & 010010 & 000000 & 001001 & 000000 & 010010 & 111111 & 110110 & 111111 & 111111 \\
000000 & 000000 & 000000 & 001001 & 000000 & 001001 & 000000 & 100100 & 111111 & 000000 & 111111 & 100100 \\
000000 & 000000 & 000000 & 000111 & 000000 & 001110 & 000000 & 101010 & 101010 & 010010 & 101010 & 001110 \\
000000 & 000000 & 000000 & 000000 & 101010 & 000111 & 000000 & 000111 & 010101 & 110001 & 111111 & 001110 \\
000000 & 000000 & 000000 & 000000 & 010101 & 001110 & 000000 & 001110 & 101010 & 100011 & 111111 & 011100 \\
000000 & 000000 & 000000 & 000000 & 000000 & 101101 & 000000 & 101101 & 000000 & 011011 & 000000 & 011011 \\
000000 & 000000 & 000000 & 000000 & 000000 & 011011 & 000000 & 011011 & 000000 & 110110 & 000000 & 110110 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 100011 & 111000 & 111111 & 011100 & 101010 & 010010 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 010010 & 100100 & 000000 & 010010 & 111111 & 011011 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 001001 & 010010 & 000000 & 001001 & 111111 & 101101 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 000111 & 110001 & 111111 & 111000 & 010101 & 100100
\end{bmatrix}$$

$$G_A^{\times} = \begin{bmatrix}
100011 & 000000 & 000000 & 000000 & 001001 & 000000 & 000000 & 000000 & 001001 & 010101 & 011100 & 010101 \\
010010 & 000000 & 000000 & 000000 & 000111 & 000111 & 000000 & 000111 & 111000 & 001110 & 101101 & 110001 \\
001001 & 000000 & 000000 & 000000 & 001001 & 001001 & 000000 & 001001 & 001001 & 101101 & 001001 & 101101 \\
000111 & 000000 & 000000 & 000000 & 000111 & 001110 & 000000 & 001110 & 111000 & 001001 & 000111 & 110110 \\
000000 & 100011 & 000000 & 000000 & 001001 & 000000 & 000000 & 111111 & 001110 & 010101 & 111000 & 000000 \\
000000 & 010010 & 000000 & 000000 & 000111 & 000111 & 000000 & 000111 & 011100 & 001110 & 011011 & 001110 \\
000000 & 001001 & 000000 & 000000 & 001001 & 001001 & 000000 & 001001 & 011011 & 101101 & 010010 & 010010 \\
000000 & 000111 & 000000 & 000000 & 000111 & 001110 & 000000 & 110001 & 110110 & 001001 & 001110 & 011100 \\
000000 & 000000 & 100011 & 000000 & 001110 & 001001 & 000000 & 011100 & 100011 & 101101 & 111000 & 000111 \\
000000 & 000000 & 010010 & 000000 & 001001 & 000000 & 000000 & 111111 & 010010 & 000000 & 100100 & 111111 \\
000000 & 000000 & 001001 & 000000 & 001110 & 000111 & 000000 & 111000 & 011100 & 110001 & 101101 & 110001 \\
000000 & 000000 & 000111 & 000000 & 001001 & 000111 & 000000 & 101101 & 101101 & 001110 & 001110 & 100100 \\
000000 & 000000 & 000000 & 100011 & 000000 & 000111 & 000000 & 010101 & 010101 & 001001 & 010101 & 000111 \\
000000 & 000000 & 000000 & 010010 & 000000 & 001001 & 000000 & 010010 & 111111 & 110110 & 111111 & 111111 \\
000000 & 000000 & 000000 & 001001 & 000000 & 001001 & 000000 & 100100 & 111111 & 000000 & 111111 & 100100 \\
000000 & 000000 & 000000 & 000111 & 000000 & 001110 & 000000 & 101010 & 101010 & 010010 & 101010 & 001110 \\
000000 & 000000 & 000000 & 000000 & 101010 & 000111 & 000000 & 000111 & 010101 & 110001 & 111111 & 001110 \\
000000 & 000000 & 000000 & 000000 & 010101 & 001110 & 000000 & 001110 & 101010 & 100011 & 111111 & 011100 \\
000000 & 000000 & 000000 & 000000 & 000000 & 101101 & 000000 & 101101 & 000000 & 011011 & 000000 & 011011 \\
000000 & 000000 & 000000 & 000000 & 000000 & 011011 & 000000 & 011011 & 000000 & 110110 & 000000 & 110110 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 100011 & 111000 & 111111 & 011100 & 101010 & 010010 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 010010 & 100100 & 000000 & 010010 & 111111 & 011011 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 001001 & 010010 & 000000 & 001001 & 111111 & 101101 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 000111 & 110001 & 111111 & 111000 & 010101 & 100100
\end{bmatrix}$$

$$G_{26}^{\times} = \begin{bmatrix}
100011 & 000000 & 000000 & 000000 & 000000 & 001001 & 000111 & 010101 & 110001 & 010101 & 110001 & 011011 \\
010010 & 000000 & 000000 & 000000 & 000000 & 000111 & 000000 & 101010 & 100100 & 000000 & 101010 & 001110 \\
001001 & 000000 & 000000 & 000000 & 000000 & 001001 & 001001 & 000000 & 001001 & 111111 & 101101 & 001001 \\
000111 & 000000 & 000000 & 000000 & 000000 & 000111 & 000111 & 000000 & 111000 & 010101 & 100100 & 000111 \\
000000 & 100011 & 000000 & 000000 & 000000 & 001001 & 000000 & 101101 & 111000 & 010010 & 011100 & 010101 \\
000000 & 010010 & 000000 & 000000 & 000000 & 000111 & 001001 & 001110 & 111111 & 100100 & 101010 & 110001 \\
000000 & 001001 & 000000 & 000000 & 000000 & 001001 & 000000 & 010010 & 010010 & 101101 & 110110 & 000000 \\
000000 & 000111 & 000000 & 000000 & 000000 & 000111 & 001001 & 110001 & 101010 & 011011 & 111111 & 011011 \\
000000 & 000000 & 100011 & 000000 & 000000 & 000000 & 000111 & 111111 & 100100 & 101101 & 110001 & 110001 \\
000000 & 000000 & 010010 & 000000 & 000000 & 000000 & 001001 & 000000 & 000000 & 011011 & 010010 & 111111 \\
000000 & 000000 & 001001 & 000000 & 000000 & 000000 & 001001 & 000000 & 110110 & 101101 & 010010 & 001001 \\
000000 & 000000 & 000111 & 000000 & 000000 & 000000 & 001110 & 111111 & 001001 & 011011 & 100011 & 100011 \\
000000 & 000000 & 000000 & 100011 & 000000 & 000000 & 001001 & 101101 & 001001 & 010101 & 000111 & 100011 \\
000000 & 000000 & 000000 & 010010 & 000000 & 000000 & 000000 & 011011 & 011011 & 111111 & 111111 & 100100 \\
000000 & 000000 & 000000 & 001001 & 000000 & 000000 & 000000 & 101101 & 101101 & 111111 & 111111 & 010010 \\
000000 & 000000 & 000000 & 000111 & 000000 & 000000 & 001001 & 011011 & 111111 & 101010 & 111000 & 101010 \\
000000 & 000000 & 000000 & 000000 & 100011 & 000111 & 001110 & 100100 & 111000 & 001110 & 110001 & 110110 \\
000000 & 000000 & 000000 & 000000 & 010010 & 001110 & 000000 & 100011 & 111111 & 110110 & 110001 & 100011 \\
000000 & 000000 & 000000 & 000000 & 001001 & 000111 & 000000 & 110001 & 111111 & 011011 & 111000 & 110001 \\
000000 & 000000 & 000000 & 000000 & 000111 & 001110 & 000111 & 001001 & 011100 & 011100 & 010101 & 000000 \\
000000 & 000000 & 000000 & 000000 & 000000 & 101010 & 001001 & 010101 & 011011 & 111111 & 111000 & 001110 \\
000000 & 000000 & 000000 & 000000 & 000000 & 010101 & 001001 & 101010 & 011011 & 111111 & 000111 & 110001 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 101101 & 000000 & 110110 & 000000 & 011011 & 110110 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 011011 & 000000 & 101101 & 000000 & 110110 & 101101
\end{bmatrix}$$

$$G_{27}^{\times} = \begin{bmatrix}
100011 & 000000 & 000000 & 000000 & 000000 & 001001 & 000111 & 010101 & 001110 & 101010 & 110001 & 011011 \\
010010 & 000000 & 000000 & 000000 & 000000 & 000111 & 000000 & 101010 & 011011 & 111111 & 101010 & 001110 \\
001001 & 000000 & 000000 & 000000 & 000000 & 001001 & 001001 & 000000 & 001001 & 111111 & 101101 & 001001 \\
000111 & 000000 & 000000 & 000000 & 000000 & 000111 & 000111 & 000000 & 111000 & 010101 & 100100 & 000111 \\
000000 & 100011 & 000000 & 000000 & 000000 & 001001 & 001001 & 010010 & 100011 & 010010 & 001110 & 110001 \\
000000 & 010010 & 000000 & 000000 & 000000 & 000111 & 001001 & 001110 & 000000 & 011011 & 101010 & 110001 \\
000000 & 001001 & 000000 & 000000 & 000000 & 001001 & 000000 & 010010 & 010010 & 101101 & 110110 & 000000 \\
000000 & 000111 & 000000 & 000000 & 000000 & 000111 & 000000 & 001110 & 001110 & 100100 & 101101 & 111111 \\
000000 & 000000 & 100011 & 000000 & 000000 & 000000 & 000111 & 111111 & 011011 & 010010 & 110001 & 110001 \\
000000 & 000000 & 010010 & 000000 & 000000 & 000000 & 001001 & 000000 & 000000 & 011011 & 010010 & 111111 \\
000000 & 000000 & 001001 & 000000 & 000000 & 000000 & 001001 & 000000 & 110110 & 101101 & 010010 & 001001 \\
000000 & 000000 & 000111 & 000000 & 000000 & 000000 & 001110 & 111111 & 110110 & 100100 & 100011 & 100011 \\
000000 & 000000 & 000000 & 100011 & 000000 & 000000 & 000000 & 010010 & 010010 & 010101 & 010101 & 000111 \\
000000 & 000000 & 000000 & 010010 & 000000 & 000000 & 000000 & 011011 & 011011 & 111111 & 111111 & 100100 \\
000000 & 000000 & 000000 & 001001 & 000000 & 000000 & 000000 & 101101 & 101101 & 111111 & 111111 & 010010 \\
000000 & 000000 & 000000 & 000111 & 000000 & 000000 & 000000 & 100100 & 100100 & 101010 & 101010 & 001110 \\
000000 & 000000 & 000000 & 000000 & 100011 & 000111 & 001110 & 100100 & 111000 & 001110 & 110001 & 110110 \\
000000 & 000000 & 000000 & 000000 & 010010 & 001110 & 000000 & 100011 & 000000 & 001001 & 110001 & 100011 \\
000000 & 000000 & 000000 & 000000 & 001001 & 000111 & 000000 & 110001 & 000000 & 100100 & 111000 & 110001 \\
000000 & 000000 & 000000 & 000000 & 000111 & 001110 & 000111 & 001001 & 011100 & 011100 & 010101 & 000000 \\
000000 & 000000 & 000000 & 000000 & 000000 & 101010 & 001001 & 010101 & 100100 & 000000 & 111000 & 001110 \\
000000 & 000000 & 000000 & 000000 & 000000 & 010101 & 001001 & 101010 & 100100 & 000000 & 000111 & 110001 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 101101 & 000000 & 110110 & 000000 & 011011 & 110110 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 011011 & 000000 & 101101 & 000000 & 110110 & 101101
\end{bmatrix}$$

$$G_{28}^{\times} = \begin{bmatrix}
100011 & 000000 & 000000 & 000000 & 000000 & 001001 & 001110 & 101010 & 101010 & 010101 & 100011 & 000000 \\
010010 & 000000 & 000000 & 000000 & 000000 & 000111 & 001001 & 101010 & 000000 & 000000 & 000111 & 010101 \\
001001 & 000000 & 000000 & 000000 & 000000 & 001001 & 001001 & 000000 & 001001 & 111111 & 101101 & 001001 \\
000111 & 000000 & 000000 & 000000 & 000000 & 000111 & 000111 & 111111 & 000111 & 010101 & 011011 & 000111 \\
000000 & 100011 & 000000 & 000000 & 000000 & 001001 & 001001 & 010010 & 100011 & 010010 & 001110 & 110001 \\
000000 & 010010 & 000000 & 000000 & 000000 & 000111 & 000000 & 001110 & 011011 & 100100 & 000111 & 101010 \\
000000 & 001001 & 000000 & 000000 & 000000 & 001001 & 000000 & 010010 & 010010 & 101101 & 110110 & 000000 \\
000000 & 000111 & 000000 & 000000 & 000000 & 000111 & 001001 & 001110 & 010101 & 011011 & 000000 & 100100 \\
000000 & 000000 & 100011 & 000000 & 000000 & 000000 & 000111 & 111111 & 100100 & 101101 & 110001 & 001110 \\
000000 & 000000 & 010010 & 000000 & 000000 & 000000 & 001001 & 000000 & 000000 & 011011 & 010010 & 111111 \\
000000 & 000000 & 001001 & 000000 & 000000 & 000000 & 001001 & 000000 & 110110 & 101101 & 010010 & 001001 \\
000000 & 000000 & 000111 & 000000 & 000000 & 000000 & 001110 & 111111 & 001001 & 011011 & 100011 & 011100 \\
000000 & 000000 & 000000 & 100011 & 000000 & 000000 & 001001 & 010010 & 110110 & 101010 & 000111 & 011100 \\
000000 & 000000 & 000000 & 010010 & 000000 & 000000 & 000000 & 011011 & 011011 & 111111 & 111111 & 100100 \\
000000 & 000000 & 000000 & 001001 & 000000 & 000000 & 000000 & 101101 & 101101 & 111111 & 111111 & 010010 \\
000000 & 000000 & 000000 & 000111 & 000000 & 000000 & 001001 & 100100 & 000000 & 010101 & 111000 & 010101 \\
000000 & 000000 & 000000 & 000000 & 100011 & 000111 & 001110 & 100100 & 111000 & 001110 & 110001 & 110110 \\
000000 & 000000 & 000000 & 000000 & 010010 & 001110 & 001001 & 100011 & 011011 & 110110 & 011100 & 111000 \\
000000 & 000000 & 000000 & 000000 & 001001 & 000111 & 001001 & 110001 & 011011 & 011011 & 010101 & 101010 \\
000000 & 000000 & 000000 & 000000 & 000111 & 001110 & 000111 & 001001 & 011100 & 011100 & 010101 & 000000 \\
000000 & 000000 & 000000 & 000000 & 000000 & 101010 & 000000 & 010101 & 111111 & 111111 & 010101 & 010101 \\
000000 & 000000 & 000000 & 000000 & 000000 & 010101 & 000000 & 101010 & 111111 & 111111 & 101010 & 101010 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 101101 & 000000 & 110110 & 000000 & 011011 & 110110 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 011011 & 000000 & 101101 & 000000 & 110110 & 101101
\end{bmatrix}$$

$$G_{29}^{\times} = \begin{bmatrix}
100011 & 000000 & 000000 & 000000 & 000000 & 000111 & 001001 & 001110 & 100100 & 110001 & 110110 & 011011 \\
010010 & 000000 & 000000 & 000000 & 000000 & 000000 & 001001 & 011100 & 000000 & 101010 & 010101 & 110001 \\
001001 & 000000 & 000000 & 000000 & 000000 & 000000 & 100100 & 001110 & 000000 & 010101 & 101010 & 111000 \\
000111 & 000000 & 000000 & 000000 & 000000 & 000111 & 011011 & 001001 & 100100 & 011011 & 100011 & 000111 \\
000000 & 100011 & 000000 & 000000 & 000000 & 000000 & 100011 & 000111 & 010010 & 101010 & 010010 & 101101 \\
000000 & 010010 & 000000 & 000000 & 000000 & 000000 & 010010 & 011011 & 011011 & 000000 & 100100 & 100100 \\
000000 & 001001 & 000000 & 000000 & 000000 & 000000 & 001001 & 010010 & 101101 & 111111 & 101101 & 101101 \\
000000 & 000111 & 000000 & 000000 & 000000 & 000000 & 000111 & 001110 & 100100 & 010101 & 100100 & 011011 \\
000000 & 000000 & 100011 & 000000 & 000000 & 000000 & 100100 & 001001 & 100011 & 011011 & 010101 & 100011 \\
000000 & 000000 & 010010 & 000000 & 000000 & 000000 & 110110 & 010010 & 010010 & 001001 & 000000 & 101101 \\
000000 & 000000 & 001001 & 000000 & 000000 & 000000 & 011011 & 001001 & 001001 & 100100 & 000000 & 110110 \\
000000 & 000000 & 000111 & 000000 & 000000 & 000000 & 001001 & 010010 & 000111 & 110110 & 101010 & 000111 \\
000000 & 000000 & 000000 & 100011 & 000000 & 000111 & 111000 & 001001 & 100011 & 001110 & 011100 & 101101 \\
000000 & 000000 & 000000 & 010010 & 000000 & 000000 & 000000 & 000111 & 100100 & 010101 & 010101 & 100011 \\
000000 & 000000 & 000000 & 001001 & 000000 & 000000 & 000000 & 011100 & 010010 & 010101 & 010101 & 001110 \\
000000 & 000000 & 000000 & 000111 & 000000 & 000111 & 111000 & 000111 & 101010 & 100100 & 110110 & 101010 \\
000000 & 000000 & 000000 & 000000 & 100011 & 000111 & 110001 & 011100 & 100011 & 111111 & 010101 & 110110 \\
000000 & 000000 & 000000 & 000000 & 010010 & 000000 & 101101 & 000111 & 100100 & 100011 & 000111 & 101010 \\
000000 & 000000 & 000000 & 000000 & 001001 & 000000 & 110110 & 011100 & 010010 & 001110 & 011100 & 101010 \\
000000 & 000000 & 000000 & 000000 & 000111 & 000111 & 101010 & 010010 & 101010 & 111000 & 011011 & 100011 \\
000000 & 000000 & 000000 & 000000 & 000000 & 100100 & 100100 & 010101 & 110110 & 100011 & 111000 & 000111 \\
000000 & 000000 & 000000 & 000000 & 000000 & 010010 & 010010 & 010101 & 011011 & 001110 & 100011 & 011100 \\
000000 & 000000 & 000000 & 000000 & 000000 & 001001 & 001001 & 010101 & 101101 & 111000 & 001110 & 110001 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 111111 & 000000 & 111111 & 111111 & 111111
\end{bmatrix}$$

$$G_{30}^{\times} = \begin{bmatrix}
100011 & 000000 & 000000 & 000000 & 000000 & 000111 & 001001 & 011011 & 100100 & 011011 & 100011 & 001110 \\
010010 & 000000 & 000000 & 000000 & 000000 & 000000 & 001001 & 011100 & 000000 & 101010 & 010101 & 110001 \\
001001 & 000000 & 000000 & 000000 & 000000 & 000000 & 100100 & 001110 & 000000 & 010101 & 101010 & 111000 \\
000111 & 000000 & 000000 & 000000 & 000000 & 000111 & 011011 & 011100 & 100100 & 110001 & 110110 & 010010 \\
000000 & 100011 & 000000 & 000000 & 000000 & 000000 & 100011 & 010010 & 010010 & 000000 & 000111 & 111000 \\
000000 & 010010 & 000000 & 000000 & 000000 & 000000 & 010010 & 011011 & 011011 & 000000 & 100100 & 100100 \\
000000 & 001001 & 000000 & 000000 & 000000 & 000000 & 001001 & 010010 & 101101 & 111111 & 101101 & 101101 \\
000000 & 000111 & 000000 & 000000 & 000000 & 000000 & 000111 & 011011 & 100100 & 111111 & 110001 & 001110 \\
000000 & 000000 & 100011 & 000000 & 000000 & 000000 & 100100 & 011100 & 011100 & 001110 & 111111 & 110110 \\
000000 & 000000 & 010010 & 000000 & 000000 & 000000 & 110110 & 010010 & 010010 & 001001 & 000000 & 101101 \\
000000 & 000000 & 001001 & 000000 & 000000 & 000000 & 011011 & 001001 & 001001 & 100100 & 000000 & 110110 \\
000000 & 000000 & 000111 & 000000 & 000000 & 000000 & 001001 & 000111 & 111000 & 100011 & 000000 & 010010 \\
000000 & 000000 & 000000 & 100011 & 000000 & 000111 & 111000 & 001001 & 011100 & 110001 & 100011 & 101101 \\
000000 & 000000 & 000000 & 010010 & 000000 & 000000 & 000000 & 000111 & 100100 & 010101 & 010101 & 100011 \\
000000 & 000000 & 000000 & 001001 & 000000 & 000000 & 000000 & 011100 & 010010 & 010101 & 010101 & 001110 \\
000000 & 000000 & 000000 & 000111 & 000000 & 000111 & 111000 & 000111 & 010101 & 011011 & 001001 & 101010 \\
000000 & 000000 & 000000 & 000000 & 100011 & 000111 & 110001 & 011100 & 100011 & 111111 & 010101 & 110110 \\
000000 & 000000 & 000000 & 000000 & 010010 & 000000 & 101101 & 000111 & 100100 & 100011 & 000111 & 101010 \\
000000 & 000000 & 000000 & 000000 & 001001 & 000000 & 110110 & 011100 & 010010 & 001110 & 011100 & 101010 \\
000000 & 000000 & 000000 & 000000 & 000111 & 000111 & 101010 & 010010 & 101010 & 111000 & 011011 & 100011 \\
000000 & 000000 & 000000 & 000000 & 000000 & 100100 & 100100 & 010101 & 110110 & 100011 & 111000 & 000111 \\
000000 & 000000 & 000000 & 000000 & 000000 & 010010 & 010010 & 010101 & 011011 & 001110 & 100011 & 011100 \\
000000 & 000000 & 000000 & 000000 & 000000 & 001001 & 001001 & 010101 & 101101 & 111000 & 001110 & 110001 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 111111 & 000000 & 111111 & 111111 & 111111
\end{bmatrix}$$

$$G_{31}^{\times} = \begin{bmatrix}
100011 & 000000 & 000000 & 000000 & 000000 & 000111 & 001001 & 001110 & 011011 & 110001 & 110110 & 100100 \\
010010 & 000000 & 000000 & 000000 & 000000 & 001001 & 000000 & 001001 & 101101 & 010010 & 011011 & 000000 \\
001001 & 000000 & 000000 & 000000 & 000000 & 001001 & 010010 & 001110 & 010010 & 111000 & 110001 & 100011 \\
000111 & 000000 & 000000 & 000000 & 000000 & 001110 & 010010 & 011100 & 110110 & 100011 & 101101 & 001001 \\
000000 & 100011 & 000000 & 000000 & 000000 & 001001 & 010101 & 010010 & 000000 & 101101 & 011100 & 100011 \\
000000 & 010010 & 000000 & 000000 & 000000 & 000000 & 010010 & 011011 & 011011 & 000000 & 100100 & 100100 \\
000000 & 001001 & 000000 & 000000 & 000000 & 000000 & 001001 & 010010 & 101101 & 111111 & 101101 & 101101 \\
000000 & 000111 & 000000 & 000000 & 000000 & 001001 & 001110 & 001110 & 001001 & 000111 & 111111 & 111111 \\
000000 & 000000 & 100011 & 000000 & 000000 & 001001 & 010010 & 001001 & 001110 & 001001 & 001110 & 111000 \\
000000 & 000000 & 010010 & 000000 & 000000 & 000000 & 001001 & 000111 & 101101 & 011100 & 010101 & 000111 \\
000000 & 000000 & 001001 & 000000 & 000000 & 000000 & 011011 & 001001 & 001001 & 100100 & 000000 & 110110 \\
000000 & 000000 & 000111 & 000000 & 000000 & 001001 & 000000 & 000111 & 010101 & 110001 & 100100 & 110110 \\
000000 & 000000 & 000000 & 100011 & 000000 & 000111 & 000111 & 011100 & 100011 & 100100 & 110110 & 000111 \\
000000 & 000000 & 000000 & 010010 & 000000 & 001001 & 001001 & 010010 & 001001 & 101101 & 011011 & 010010 \\
000000 & 000000 & 000000 & 001001 & 000000 & 001001 & 001001 & 001001 & 111111 & 101101 & 011011 & 111111 \\
000000 & 000000 & 000000 & 000111 & 000000 & 001110 & 001110 & 000111 & 000111 & 110110 & 010010 & 110001 \\
000000 & 000000 & 000000 & 000000 & 100011 & 001110 & 000111 & 011100 & 110001 & 010010 & 001110 & 101101 \\
000000 & 000000 & 000000 & 000000 & 010010 & 001001 & 011011 & 000111 & 110110 & 001110 & 011100 & 110001 \\
000000 & 000000 & 000000 & 000000 & 001001 & 001001 & 000000 & 011100 & 000000 & 100011 & 000111 & 110001 \\
000000 & 000000 & 000000 & 000000 & 000111 & 000111 & 010101 & 000111 & 010101 & 101101 & 001110 & 001001 \\
000000 & 000000 & 000000 & 000000 & 000000 & 101101 & 010010 & 010101 & 100100 & 001110 & 100011 & 011100 \\
000000 & 000000 & 000000 & 000000 & 000000 & 011011 & 011011 & 000000 & 110110 & 110110 & 101101 & 101101 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 111111 & 010101 & 111111 & 010101 & 010101 & 101010 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 111111 & 000000 & 111111 & 111111 & 111111
\end{bmatrix}$$

$$G_{32}^{\times} = \begin{bmatrix}
100011 & 000000 & 000000 & 000111 & 001001 & 000000 & 000111 & 000000 & 100011 & 100011 & 101101 & 111000 \\
010010 & 000000 & 000000 & 001110 & 000111 & 000000 & 000000 & 000000 & 010101 & 101101 & 101010 & 111111 \\
001001 & 000000 & 000000 & 000111 & 001001 & 000111 & 000000 & 000000 & 110110 & 011100 & 011100 & 011011 \\
000111 & 000000 & 000000 & 001110 & 000111 & 000111 & 000000 & 001110 & 110001 & 111000 & 101010 & 110110 \\
000000 & 100011 & 000000 & 000111 & 001110 & 000111 & 000000 & 000000 & 011100 & 111000 & 011100 & 001110 \\
000000 & 010010 & 000000 & 001110 & 001001 & 000111 & 000111 & 001110 & 000111 & 011011 & 010010 & 001110 \\
000000 & 001001 & 000000 & 000111 & 001110 & 000000 & 000000 & 001001 & 110001 & 101101 & 111000 & 011011 \\
000000 & 000111 & 000000 & 001110 & 001001 & 000000 & 000000 & 000000 & 000000 & 100100 & 101010 & 010101 \\
000000 & 000000 & 100011 & 001001 & 001001 & 000111 & 000111 & 001001 & 100100 & 000111 & 001001 & 010010 \\
000000 & 000000 & 010010 & 000111 & 000111 & 000000 & 000111 & 000111 & 000000 & 001110 & 110110 & 110001 \\
000000 & 000000 & 001001 & 001001 & 001001 & 000000 & 000111 & 001110 & 110001 & 111000 & 110001 & 010101 \\
000000 & 000000 & 000111 & 000111 & 000111 & 000111 & 000000 & 001110 & 011011 & 100100 & 000000 & 100011 \\
000000 & 000000 & 000000 & 101010 & 000000 & 000111 & 000000 & 001001 & 010010 & 101010 & 110001 & 000000 \\
000000 & 000000 & 000000 & 010101 & 000000 & 000111 & 000111 & 001110 & 111000 & 000000 & 010010 & 110001 \\
000000 & 000000 & 000000 & 000000 & 101010 & 000111 & 000000 & 000000 & 011100 & 101010 & 001001 & 100100 \\
000000 & 000000 & 000000 & 000000 & 010101 & 000111 & 000111 & 000111 & 001001 & 000000 & 010101 & 010101 \\
000000 & 000000 & 000000 & 000000 & 000000 & 100100 & 000111 & 000111 & 000111 & 010101 & 110001 & 000111 \\
000000 & 000000 & 000000 & 000000 & 000000 & 010010 & 000111 & 000111 & 011100 & 010101 & 101010 & 101010 \\
000000 & 000000 & 000000 & 000000 & 000000 & 001001 & 000111 & 000111 & 110001 & 010101 & 000111 & 011100 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 100100 & 001001 & 001001 & 111111 & 001001 & 111111 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 010010 & 001001 & 010010 & 111111 & 111111 & 001001 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 001001 & 001001 & 111111 & 111111 & 100100 & 010010 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 101101 & 110110 & 000000 & 011011 & 110110 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 011011 & 101101 & 000000 & 110110 & 101101
\end{bmatrix}$$

$$G_{33}^{\times} = \begin{bmatrix}
100011 & 000000 & 000000 & 000111 & 001001 & 000000 & 000111 & 000000 & 001110 & 100011 & 101101 & 111000 \\
010010 & 000000 & 000000 & 001110 & 000111 & 000000 & 000000 & 000000 & 010101 & 101101 & 101010 & 111111 \\
001001 & 000000 & 000000 & 000111 & 001001 & 000111 & 000000 & 000000 & 011011 & 011100 & 110001 & 110110 \\
000111 & 000000 & 000000 & 001110 & 000111 & 000111 & 000000 & 001110 & 000111 & 111000 & 011100 & 000000 \\
000000 & 100011 & 000000 & 000111 & 001110 & 000111 & 000000 & 000000 & 110001 & 111000 & 110001 & 100011 \\
000000 & 010010 & 000000 & 001110 & 001001 & 000111 & 000111 & 001110 & 011100 & 011011 & 100100 & 111000 \\
000000 & 001001 & 000000 & 000111 & 001110 & 000000 & 000000 & 001001 & 000111 & 101101 & 001110 & 101101 \\
000000 & 000111 & 000000 & 001110 & 001001 & 000000 & 000000 & 000000 & 000000 & 100100 & 101010 & 010101 \\
000000 & 000000 & 100011 & 001001 & 001001 & 000111 & 000111 & 001001 & 010010 & 000111 & 010010 & 001001 \\
000000 & 000000 & 010010 & 000111 & 000111 & 000000 & 000111 & 000111 & 000000 & 001110 & 011011 & 011100 \\
000000 & 000000 & 001001 & 001001 & 001001 & 000000 & 000111 & 001110 & 000111 & 111000 & 101010 & 001110 \\
000000 & 000000 & 000111 & 000111 & 000111 & 000111 & 000000 & 001110 & 101101 & 100100 & 110110 & 010101 \\
000000 & 000000 & 000000 & 101010 & 000000 & 000111 & 000000 & 001001 & 001001 & 101010 & 101010 & 011011 \\
000000 & 000000 & 000000 & 010101 & 000000 & 000111 & 000111 & 001110 & 100011 & 000000 & 100100 & 000111 \\
000000 & 000000 & 000000 & 000000 & 101010 & 000111 & 000000 & 000000 & 110001 & 101010 & 100100 & 001001 \\
000000 & 000000 & 000000 & 000000 & 010101 & 000111 & 000111 & 000111 & 100100 & 000000 & 010101 & 010101 \\
000000 & 000000 & 000000 & 000000 & 000000 & 100100 & 000111 & 000111 & 011100 & 010101 & 000111 & 110001 \\
000000 & 000000 & 000000 & 000000 & 000000 & 010010 & 000111 & 000111 & 110001 & 010101 & 101010 & 101010 \\
000000 & 000000 & 000000 & 000000 & 000000 & 001001 & 000111 & 000111 & 000111 & 010101 & 011100 & 000111 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 100100 & 001001 & 100100 & 111111 & 111111 & 001001 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 010010 & 001001 & 001001 & 111111 & 001001 & 111111 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 001001 & 001001 & 111111 & 111111 & 010010 & 100100 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 101101 & 011011 & 000000 & 110110 & 011011 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 011011 & 110110 & 000000 & 101101 & 110110
\end{bmatrix}$$

$$G_{34}^{\times} = \begin{bmatrix}
100011 & 000000 & 000000 & 000000 & 011100 & 000111 & 000000 & 000111 & 100100 & 010010 & 011011 & 101101 \\
010010 & 000000 & 000000 & 000000 & 010010 & 000000 & 001001 & 000000 & 101101 & 011011 & 000000 & 010010 \\
001001 & 000000 & 000000 & 000000 & 001001 & 000000 & 001110 & 001001 & 101010 & 010010 & 011100 & 101010 \\
000111 & 000000 & 000000 & 000000 & 000111 & 000000 & 001001 & 000000 & 111000 & 100100 & 101010 & 101101 \\
000000 & 100011 & 000000 & 000000 & 010101 & 000111 & 001110 & 000000 & 000000 & 110001 & 001110 & 100100 \\
000000 & 010010 & 000000 & 000000 & 000000 & 000111 & 001110 & 000111 & 010010 & 110110 & 000000 & 101010 \\
000000 & 001001 & 000000 & 000000 & 000000 & 000111 & 001110 & 000111 & 010010 & 011011 & 110110 & 011100 \\
000000 & 000111 & 000000 & 000000 & 010101 & 000000 & 000000 & 000111 & 010010 & 011100 & 100011 & 100011 \\
000000 & 000000 & 100011 & 000000 & 001001 & 000000 & 000000 & 001001 & 100011 & 101010 & 101010 & 000000 \\
000000 & 000000 & 010010 & 000000 & 010010 & 000111 & 001110 & 001110 & 011011 & 000000 & 000000 & 010101 \\
000000 & 000000 & 001001 & 000000 & 001001 & 000111 & 001110 & 001110 & 011011 & 000000 & 011011 & 001110 \\
000000 & 000000 & 000111 & 000000 & 010010 & 000000 & 000000 & 001001 & 011100 & 010101 & 001110 & 011011 \\
000000 & 000000 & 000000 & 100011 & 000111 & 000111 & 000111 & 000000 & 000111 & 001110 & 100100 & 110110 \\
000000 & 000000 & 000000 & 010010 & 011011 & 000111 & 001001 & 001110 & 101010 & 001001 & 100011 & 010010 \\
000000 & 000000 & 000000 & 001001 & 010010 & 000000 & 001001 & 001001 & 011011 & 100100 & 100100 & 110110 \\
000000 & 000000 & 000000 & 000111 & 001110 & 000111 & 001001 & 000000 & 011011 & 100011 & 000111 & 100011 \\
000000 & 000000 & 000000 & 000000 & 111111 & 000111 & 001110 & 001110 & 100100 & 111111 & 010010 & 111000 \\
000000 & 000000 & 000000 & 000000 & 000000 & 100100 & 001001 & 001001 & 110110 & 000000 & 011011 & 100100 \\
000000 & 000000 & 000000 & 000000 & 000000 & 010010 & 001110 & 000000 & 101010 & 111111 & 011100 & 011100 \\
000000 & 000000 & 000000 & 000000 & 000000 & 001001 & 000111 & 000000 & 010101 & 111111 & 001110 & 001110 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 101010 & 001001 & 011100 & 111111 & 011100 & 100011 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 010101 & 001001 & 100011 & 111111 & 100011 & 011100 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 101101 & 101101 & 000000 & 101101 & 101101 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 011011 & 011011 & 000000 & 011011 & 011011
\end{bmatrix}$$

$$G_{35}^{\times} = \begin{bmatrix}
100011 & 000000 & 000000 & 000000 & 011100 & 000111 & 000000 & 000111 & 011011 & 010010 & 011011 & 010010 \\
010010 & 000000 & 000000 & 000000 & 010010 & 000000 & 001001 & 000000 & 101101 & 011011 & 000000 & 010010 \\
001001 & 000000 & 000000 & 000000 & 001001 & 000000 & 001110 & 001001 & 101010 & 010010 & 011100 & 101010 \\
000111 & 000000 & 000000 & 000000 & 000111 & 000000 & 001001 & 001001 & 110001 & 011011 & 100011 & 011011 \\
000000 & 100011 & 000000 & 000000 & 010101 & 000111 & 001110 & 000000 & 000000 & 110001 & 001110 & 100100 \\
000000 & 010010 & 000000 & 000000 & 000000 & 000111 & 001110 & 001110 & 100100 & 001001 & 001001 & 100011 \\
000000 & 001001 & 000000 & 000000 & 000000 & 000111 & 001110 & 001110 & 100100 & 100100 & 111111 & 010101 \\
000000 & 000111 & 000000 & 000000 & 010101 & 000000 & 000000 & 001110 & 100100 & 100011 & 101010 & 101010 \\
000000 & 000000 & 100011 & 000000 & 001001 & 000000 & 000000 & 000000 & 010101 & 010101 & 100011 & 001001 \\
000000 & 000000 & 010010 & 000000 & 010010 & 000111 & 001110 & 000111 & 101101 & 111111 & 001001 & 011100 \\
000000 & 000000 & 001001 & 000000 & 001001 & 000111 & 001110 & 000111 & 101101 & 111111 & 010010 & 000111 \\
000000 & 000000 & 000111 & 000000 & 010010 & 000000 & 000000 & 000000 & 101010 & 101010 & 000111 & 010010 \\
000000 & 000000 & 000000 & 100011 & 000111 & 000000 & 001001 & 000111 & 010101 & 001110 & 000000 & 111000 \\
000000 & 000000 & 000000 & 010010 & 011011 & 000111 & 001001 & 000111 & 011100 & 110110 & 101010 & 011011 \\
000000 & 000000 & 000000 & 001001 & 010010 & 000000 & 001001 & 001001 & 011011 & 100100 & 100100 & 110110 \\
000000 & 000000 & 000000 & 000111 & 001110 & 000000 & 000111 & 000111 & 001001 & 100011 & 100011 & 101101 \\
000000 & 000000 & 000000 & 000000 & 111111 & 000111 & 001110 & 000111 & 010010 & 000000 & 011011 & 110001 \\
000000 & 000000 & 000000 & 000000 & 000000 & 100100 & 001001 & 001001 & 110110 & 000000 & 011011 & 100100 \\
000000 & 000000 & 000000 & 000000 & 000000 & 010010 & 001110 & 000000 & 101010 & 111111 & 011100 & 011100 \\
000000 & 000000 & 000000 & 000000 & 000000 & 001001 & 000111 & 000000 & 010101 & 111111 & 001110 & 001110 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 101010 & 001001 & 011100 & 111111 & 011100 & 100011 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 010101 & 001001 & 100011 & 111111 & 100011 & 011100 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 101101 & 101101 & 000000 & 101101 & 101101 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 000000 & 011011 & 011011 & 000000 & 011011 & 011011
\end{bmatrix}$$

$$G_{36}^{\times} = \begin{bmatrix}
100011 & 000000 & 000000 & 000000 & 000000 & 000000 & 101101 & 111000 & 100100 & 000111 & 100011 & 111111 \\
010010 & 000000 & 000000 & 000000 & 000000 & 000000 & 011011 & 100100 & 110110 & 100100 & 010010 & 000000 \\
001001 & 000000 & 000000 & 000000 & 000000 & 000000 & 101101 & 010010 & 011011 & 010010 & 001001 & 000000 \\
000111 & 000000 & 000000 & 000000 & 000000 & 000000 & 011011 & 110001 & 001001 & 001110 & 000111 & 111111 \\
000000 & 100011 & 000000 & 000000 & 000000 & 000000 & 100011 & 011100 & 010101 & 000111 & 001001 & 110001 \\
000000 & 010010 & 000000 & 000000 & 000000 & 000000 & 010010 & 010010 & 111111 & 100100 & 101101 & 001001 \\
000000 & 001001 & 000000 & 000000 & 000000 & 000000 & 001001 & 001001 & 111111 & 010010 & 110110 & 100100 \\
000000 & 000111 & 000000 & 000000 & 000000 & 000000 & 000111 & 111000 & 101010 & 001110 & 010010 & 100011 \\
000000 & 000000 & 100011 & 000000 & 000000 & 000000 & 111111 & 001001 & 111000 & 001110 & 000111 & 011011 \\
000000 & 000000 & 010010 & 000000 & 000000 & 000000 & 000000 & 101101 & 100100 & 001001 & 100100 & 110110 \\
000000 & 000000 & 001001 & 000000 & 000000 & 000000 & 000000 & 110110 & 010010 & 100100 & 010010 & 011011 \\
000000 & 000000 & 000111 & 000000 & 000000 & 000000 & 111111 & 010010 & 110001 & 011100 & 001110 & 110110 \\
000000 & 000000 & 000000 & 100011 & 000000 & 000000 & 110001 & 101010 & 100011 & 001001 & 001110 & 111000 \\
000000 & 000000 & 000000 & 010010 & 000000 & 000000 & 001001 & 111111 & 010010 & 101101 & 001001 & 100100 \\
000000 & 000000 & 000000 & 001001 & 000000 & 000000 & 100100 & 111111 & 001001 & 110110 & 100100 & 010010 \\
000000 & 000000 & 000000 & 000111 & 000000 & 000000 & 100011 & 010101 & 000111 & 010010 & 011100 & 110001 \\
000000 & 000000 & 000000 & 000000 & 100011 & 000000 & 111000 & 111111 & 000000 & 010101 & 100011 & 010010 \\
000000 & 000000 & 000000 & 000000 & 010010 & 000000 & 100100 & 000000 & 000000 & 111111 & 010010 & 011011 \\
000000 & 000000 & 000000 & 000000 & 001001 & 000000 & 010010 & 000000 & 000000 & 111111 & 001001 & 101101 \\
000000 & 000000 & 000000 & 000000 & 000111 & 000000 & 110001 & 111111 & 000000 & 101010 & 000111 & 100100 \\
000000 & 000000 & 000000 & 000000 & 000000 & 100011 & 010010 & 000000 & 111111 & 000111 & 101010 & 111000 \\
000000 & 000000 & 000000 & 000000 & 000000 & 010010 & 011011 & 000000 & 000000 & 100100 & 111111 & 100100 \\
000000 & 000000 & 000000 & 000000 & 000000 & 001001 & 101101 & 000000 & 000000 & 010010 & 111111 & 010010 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000111 & 100100 & 000000 & 111111 & 001110 & 010101 & 110001
\end{bmatrix}$$

$$G_B^{\times} = \left[\begin{array}{cccccccccccc}
100011 & 000000 & 000000 & 000000 & 000000 & 000000 & 010010 & 111000 & 011011 & 111000 & 011100 & 111111 \\
010010 & 000000 & 000000 & 000000 & 000000 & 000000 & 011011 & 100100 & 110110 & 100100 & 010010 & 000000 \\
001001 & 000000 & 000000 & 000000 & 000000 & 000000 & 101101 & 010010 & 011011 & 010010 & 001001 & 000000 \\
000111 & 000000 & 000000 & 000000 & 000000 & 000000 & 100100 & 110001 & 110110 & 110001 & 111000 & 111111 \\
000000 & 100011 & 000000 & 000000 & 000000 & 000000 & 100011 & 011100 & 101010 & 111000 & 001001 & 001110 \\
000000 & 010010 & 000000 & 000000 & 000000 & 000000 & 010010 & 010010 & 111111 & 100100 & 101101 & 001001 \\
000000 & 001001 & 000000 & 000000 & 000000 & 000000 & 001001 & 001001 & 111111 & 010010 & 110110 & 100100 \\
000000 & 000111 & 000000 & 000000 & 000000 & 000000 & 000111 & 111000 & 010101 & 110001 & 010010 & 011100 \\
000000 & 000000 & 100011 & 000000 & 000000 & 000000 & 000000 & 001001 & 111000 & 001110 & 111000 & 100100 \\
000000 & 000000 & 010010 & 000000 & 000000 & 000000 & 000000 & 101101 & 100100 & 001001 & 100100 & 110110 \\
000000 & 000000 & 001001 & 000000 & 000000 & 000000 & 000000 & 110110 & 010010 & 100100 & 010010 & 011011 \\
000000 & 000000 & 000111 & 000000 & 000000 & 000000 & 000000 & 010010 & 110001 & 011100 & 110001 & 001001 \\
000000 & 000000 & 000000 & 100011 & 000000 & 000000 & 001110 & 101010 & 100011 & 001001 & 110001 & 000111 \\
000000 & 000000 & 000000 & 010010 & 000000 & 000000 & 001001 & 111111 & 010010 & 101101 & 001001 & 100100 \\
000000 & 000000 & 000000 & 001001 & 000000 & 000000 & 100100 & 111111 & 001001 & 110110 & 100100 & 010010 \\
000000 & 000000 & 000000 & 000111 & 000000 & 000000 & 011100 & 010101 & 000111 & 010010 & 100011 & 001110 \\
000000 & 000000 & 000000 & 000000 & 100011 & 000000 & 111000 & 111111 & 111111 & 101010 & 100011 & 101101 \\
000000 & 000000 & 000000 & 000000 & 010010 & 000000 & 100100 & 000000 & 000000 & 111111 & 010010 & 011011 \\
000000 & 000000 & 000000 & 000000 & 001001 & 000000 & 010010 & 000000 & 000000 & 111111 & 001001 & 101101 \\
000000 & 000000 & 000000 & 000000 & 000111 & 000000 & 110001 & 111111 & 111111 & 010101 & 000111 & 011011 \\
000000 & 000000 & 000000 & 000000 & 000000 & 100011 & 010010 & 000000 & 111111 & 000111 & 101010 & 111000 \\
000000 & 000000 & 000000 & 000000 & 000000 & 010010 & 011011 & 000000 & 000000 & 100100 & 111111 & 100100 \\
000000 & 000000 & 000000 & 000000 & 000000 & 001001 & 101101 & 000000 & 000000 & 010010 & 111111 & 010010 \\
000000 & 000000 & 000000 & 000000 & 000000 & 000111 & 100100 & 000000 & 111111 & 001110 & 010101 & 110001
\end{array}\right]$$