Brock University

Department of Computer Science

The Extended Quadratic Residue Code is the only (48; 24; 12) Self-Dual Doubly-Even Code

Sheridan K. Houghten, C.W.H. Lam, L.H. Thiel, and J.A. Parker
Technical Report # CS-02-09
May 2002

Brock University
Department of Computer Science
St. Catharines, Ontario
Canada L2S 3A1
www.cosc.brocku.ca

The Extended Quadratic Residue Code is the only $(48, 24, 12)$ Self-Dual Doubly-Even Code

S.K. Houghten*
Department of Computer Science
Brock University
St. Catharines, Ontario, Canada L2S 3A1,
and
C.W.H. Lam †, L.H. Thiel, and J.A. Parker
Department of Computer Science
Concordia University
Montreal, Québec, Canada H3G 1M8

Abstract

The largest minimum weight of a self-dual doubly-even binary (n, k, d) code is $d = 4\lfloor n/24 \rfloor + 4$. Of such codes with length divisible by 24, the Golay Code is the only $(24, 12, 8)$ code, the Extended Quadratic Residue Code is the only known $(48, 24, 12)$ code, and there is no known $(72, 36, 16)$ code. One may partition the search for a $(48, 24, 12)$ self-dual doubly-even code into 3 cases. A previous search assuming one of the cases found only the Extended Quadratic Residue Code. In this paper we examine the remaining 2 cases. Separate searches assuming each of the remaining cases found no codes and thus the Extended Quadratic Residue Code is the only doubly-even self-dual $(48, 24, 12)$ code.

1 Introduction

In this paper we shall use the standard definitions and terminology in coding theory, which can be found in [3, 14, 19].

The largest minimum weight of a self-dual, doubly-even $(n, n/2, d)$ code over $GF(2)$ is $d = 4\lfloor n/24 \rfloor + 4$ (see [15]). For $n \geq 74$, there is a further upper bound on minimum weight, namely $d \leq 2\lfloor (n + 6)/10 \rfloor$ (see [7]). A self-dual code that has the largest possible minimum weight is an *extremal code*.

Binary self-dual codes of length up to 32 are classified in [5] and [6]. All of these have a non-trivial automorphism group. However, in [17], it is shown that most binary self-dual codes have only a trivial automorphism group.

*Supported in part by the Natural Sciences and Engineering Research Council of Canada
†Supported in part by the Natural Sciences and Engineering Research Council of Canada

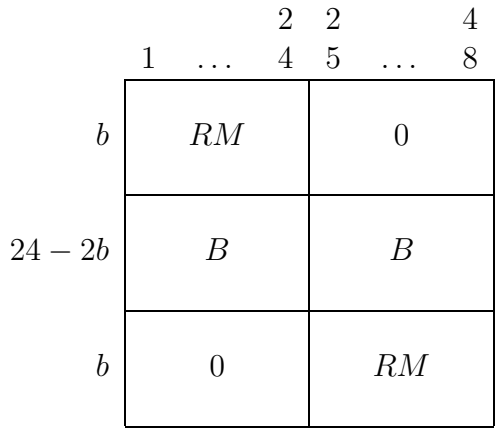

Figure 1: Organization of Generator Matrix for C_{48}

Extremal self-dual, doubly-even codes with length divisible by 24 are of particular interest because for any non-zero weight w, the codewords of weight w form a 5-design [1]. Of such codes, the Golay code is the only $(24, 12, 8)$ code and the Extended Quadratic Residue Code, which we shall call QR, is the only known $(48, 24, 12)$ code.

In [11], it is shown that any extremal self-dual doubly-even code C of length 48 with a nontrivial automorphism of odd order is equivalent to the Extended Quadratic Residue Code.

The following is the main result in this paper.

Theorem 1 *The Extended Quadratic Residue Code is the only* $(48, 24, 12)$ *self-dual doubly-even code.*

The proof is based on a exhaustive search for a generator matrix of such a code. The search is partitioned into 3 cases, namely $b = 2$, 3 or 4, where b is the maximum dimension of a subcode of length 24. A search assuming the case $b = 4$ has previously been completed, finding only the Extended Quadratic Residue Code [8, 10]. We note that in [13], the author calculates by hand that no new codes are possible if $b = 2$. Thus, our computer search is an independent verification of this result. We also note that Hans Georg Schaathun [21] has found a possibly more efficient search strategy based on using sets of higher weights.

As for the next case, there is no known self-dual doubly-even $(72, 36, 16)$ code [22], but [4] finds all the odd primes p which can divide the order of the group of such a code if it exists. These are $p = 23$, 17, 11, 7, 5 and 3. In [18], [20], and [12], the cases $p = 23$, 17 and 11 are eliminated respectively. With Schaathun's improved strategy, a computer search for this code may be possible.

2 Organization of the Generator Matrix

As shown in [8, 10], the generator matrix C_{48} can be organized as shown in Fig. 1, where RM is a $(24, b, 12)$ code obtained by taking $3 \cdot 2^{4-b}$ copies of the 1st-order Reed-Muller code $R(1, b-1)$. The code B, together with the code RM, form the dual of RM, therefore all vectors in B are chosen from $RM^{\perp}$.

In the same work it is shown that $2 \leq b \leq 4$. One may therefore divide the search into 3 cases, namely $b = 2$, $b = 3$ and $b = 4$. The results of a search assuming $b = 4$ are reported in [8, 10]. In this search, only codes isomorphic to the Extended Quadratic Residue code QR were found.

In this paper, we examine the two remaining cases, namely $b = 2$ and $b = 3$.

3 Organization of Generator Matrix for $b = 2$

Let C_{48} be a $(48, 24, 12)$ self-dual doubly-even code in which the generator matrix is organized as shown in Fig. 1, with $b = 2$.

In this case, the weight enumerator of RM is:

$$\begin{aligned} a_0 &= a_{24} = 1, \\ a_{12} &= \quad\ 2, \end{aligned}$$

where a_i is the number of weight-i codewords.

From the MacWilliams Identity, we can obtain the weight enumerator of $RM^{\perp}$, namely:

$$\begin{aligned} a_0 &= a_{24} = 1, \\ a_2 &= a_{22} = 132, \\ a_4 &= a_{20} = 5\ 346, \\ a_6 &= a_{18} = 67\ 188, \\ a_8 &= a_{16} = 367\ 983, \\ a_{10} &= a_{14} = 980\ 232, \\ a_{12} &= \quad\ 1\ 352\ 540. \end{aligned}$$

Let M be the generator matrix of C_{48}.

Two sections of M remain to be filled in. The first of these sections is the subspace generated by rows 3–22, restricted to their first 24 columns. Call these restricted rows row_i(left), $3 \leq i \leq 22$. The second of these sections is the subspace generated by rows 3–22, restricted to their last 24 columns. Call these restricted rows row_i(right), $3 \leq i \leq 22$.

Note that in both of these sections, all vectors are in $RM^{\perp}$. Therefore in particular, row_i(left) and row_i(right) must both be chosen from $RM^{\perp}$, for $3 \leq i \leq 22$.

Because the number of weight-12 codewords in C_{48} is non-zero, we use as many of these as possible when trying to complete the remainder of M. This use of weight-12 codewords is aimed at reducing the number of possibilities we must consider for each row.

Similarly, we would like to reduce the number of possibilities for row_i(left) and row_i(right). Therefore we shall attempt to complete the remainder of M using weight-10 words from $RM^{\perp}$ for row_i(left), and weight-2 words from $RM^{\perp}$ for row_i(right).

In fact, we may assume that M has the form shown in Fig. 2. The reasoning is as follows.

111111111111	000000000000	000000000000	000000000000
000000000000	111111111111	000000000000	000000000000
		110000000000	
		101000000000	
		100100000000	
		100010000000	
wt	10	100001000000	
		100000100000	
		100000010000	
		100000001000	
		100000000100	
		100000000010	
			110000000000
			101000000000
			100100000000
			100010000000
wt	10		100001000000
			100000100000
			100000010000
			100000001000
			100000000100
			100000000010
000000000000	000000000000	111111111111	000000000000
000000000000	000000000000	000000000000	111111111111

Figure 2: Structure of Generator Matrix for $b = 2$

3.1 Configuration on the Right-Hand Side

Let us consider $row_i(\text{right})$, $3 \leq i \leq 22$. Based on the configuration in the last two rows, we may divide $row_i(\text{right})$ into 2 column-blocks, each of length 12. There are 132 codewords of weight 2 in $RM^{\perp}$, and we shall attempt to use these to complete $row_i(\text{right})$, $3 \leq i \leq 22$. If we place a single one in each of the two column-blocks of $row_i(\text{right})$, then $row_i + row_{23} + row_{24}$ is a vector of weight 22, which is not doubly-even. Therefore we must place both ones in the same column-block. There are $\begin{pmatrix} 12 \\ 2 \end{pmatrix} = 66$ ways to place 2 ones in a column-block, and thus 66 codewords of weight 2 in $RM^{\perp}$ may be obtained from each column-block. Exactly 11 of these 66 codewords are linearly independent. All 66 codewords in the first column-block may be generated by $row_3(\text{right}), \ldots, row_{12}(\text{right})$ and $row_{23}(\text{right})$. All 66 codewords in the second column-block may be generated by $row_{13}(\text{right}), \ldots, row_{22}(\text{right})$ and $row_{24}(\text{right})$. All of the required 132 codewords of weight 2 in $RM^{\perp}$ may thus be obtained from the 2 column-blocks. One may now verify that $RM^{\perp}$ is generated by $row_3(\text{right}), \ldots, row_{24}(\text{right})$.

3.2 Configuration on the Left-Hand Side

Observe that we have divided the remaining incomplete rows of the generator matrix into 2 row-blocks. These are $\{row_3, \ldots, row_{12}\}$, and $\{row_{13}, \ldots, row_{22}\}$.

Now consider $row_i(\text{left})$, $3 \leq i \leq 22$. Based on the configuration in the first two rows, we may divide $row_i(\text{left})$ into 2 column-blocks, each of length 12. Since $wt(row_i(\text{right})) = 2$, and the minimum weight of C_{48} is 12, then $wt(row_i(\text{left})) \geq 12 - 2 = 10$. Furthermore, if $wt(row_i(\text{left})) > 14$ then $wt(row_i + row_1 + row_2) < 12$. Therefore $wt(row_i(\text{left})) = 10$ or 14. However if $wt(row_i(\text{left})) = 14$ then $wt(row_i(\text{left}) + row_1(\text{left}) + row_2(\text{left})) = 10$. Therefore we may assume that $wt(row_i(\text{left})) = 10$ for $3 \leq i \leq 22$.

After $row_3(\text{left}), \ldots, row_{22}(\text{left})$ are filled in, one may verify that $RM^{\perp}$ is generated by $row_1(\text{left}), \ldots, row_{22}(\text{left})$.

Let us further consider $row_i(\text{left}), 3 \leq i \leq 22$. Let b_j be the number of column-blocks in $row_i(\text{left})$ which have j ones. Clearly $0 \leq j \leq 10$. Since $wt(row_i(\text{left})) = 10$, we have $\sum_{j=0}^{10} j \cdot b_j = 10$.

Suppose there is a column-block of j ones, $j \geq 7$ in $row_i(\text{left})$. Then the other column-block of $row_i(\text{left})$ must contain $10 - j$ ones. If the column-block of j ones is the first block of $row_i(\text{left})$, then add row_i to row_1. Otherwise, add row_i to row_2. The result of the addition is a vector of weight $((12 - j) + (10 - j) + 2) = 24 - 2j$ which is ≤ 10 when $j \geq 7$. This is less than the minimum weight of the code, and therefore $b_j = 0$ for $j \geq 7$.

Now suppose that there is a column-block of j ones, $j \leq 3$ in $row_i(\text{left})$. If the column-block of j ones is the first column-block of $row_i(\text{left})$, then add row_i to row_2. Otherwise, add row_i to row_1. The result of the addition is a vector of weight $(j) + (12 - (10 - j)) + 2 = 4 + 2j$, which is ≤ 10 when $j \leq 3$. Therefore $b_j = 0$ for $j \leq 3$.

We now have two cases to consider. First consider the case in which there are 2

column-blocks of 5 ones in row_i(left). By adding row_i to row_1, we obtain a vector of weight $(12-5)+(5)+2=14$, which is not doubly-even. Therefore there must be one column-block of 6 ones, and one column-block of 4 ones, in row_i(left).

This concludes our examination of the general organization of the generator matrix for the case in which RM has $b=2$ rows. We shall now consider the remaining case, namely $b=3$.

4 Organization of Generator Matrix for $b=3$

Let C_{48} be a $(48,24,12)$ self-dual doubly-even code in which the generator matrix is organized as shown in Fig. 1, with $b=3$.

In this case, the weight enumerator of RM is:

$$\begin{aligned} a_0 &= a_{24} = 1, \\ a_{12} &= 6. \end{aligned}$$

From the MacWilliams Identity, we can obtain the weight enumerator of $RM^{\perp}$, namely:

$$\begin{aligned} a_0 &= a_{24} = 1, \\ a_2 &= a_{22} = 60, \\ a_4 &= a_{20} = 2\ 706, \\ a_6 &= a_{18} = 33\ 484, \\ a_8 &= a_{16} = 184\ 239, \\ a_{10} &= a_{14} = 489\ 720, \\ a_{12} &= 676\ 732. \end{aligned}$$

We shall use the same notation as for $b=2$. Namely, let M be the generator matrix of C_{48}. As for $b=2$, two sections of M remain to be filled in. The first of these sections is the subspace generated by rows 4–21, restricted to their first 24 columns. Call these restricted rows row_i(left), $4 \le i \le 21$. The second of these sections is the subspace generated by rows 4–21, restricted to their last 24 columns. Call these restricted rows row_i(right), $4 \le i \le 21$.

As before, row_i(left) and row_i(right) must both be chosen from $RM^{\perp}$, for $4 \le i \le 21$. We would again like to complete the remainder of M using weight-10 words from $RM^{\perp}$ for row_i(left), and weight-2 words from $RM^{\perp}$ for row_i(right). However, as we shall see, it is not possible in this case. In fact, there must be one row composed of a weight-8 or weight-12 word from $RM^{\perp}$ for row_i(left), and a weight-4 word from $RM^{\perp}$ for row_i(right). Furthermore we may assume that M has the form shown in Fig. 3, as explained below.

4.1 Configuration on the Right-Hand Side

Let us consider row_i(right), $4 \le i \le 21$. Based on the configuration in the last three rows, we may divide row_i(right) into 4 column-blocks, each of length 6. There are 60 codewords of weight 2 in $RM^{\perp}$, and we shall attempt to use these to complete row_i(right), $4 \le i \le 21$. If we place a single one in separate column-blocks of

<table>
<tr><td>111111
000000
111111</td><td>111111
000000
000000</td><td>000000
111111
111111</td><td>000000
111111
000000</td><td>000000
000000
000000</td><td>000000
000000
000000</td><td>000000
000000
000000</td><td>000000
000000
000000</td></tr>
<tr><td colspan="4">wt 10</td><td>110000
101000
100100
100010
100001</td><td></td><td></td><td></td></tr>
<tr><td colspan="4">wt 10</td><td></td><td>110000
101000
100100
100010</td><td></td><td></td></tr>
<tr><td colspan="4">wt 10</td><td></td><td></td><td>110000
101000
100100
100010</td><td></td></tr>
<tr><td colspan="4">wt 10</td><td></td><td></td><td></td><td>110000
101000
100100
100010</td></tr>
<tr><td></td><td>wt</td><td>8 or 12</td><td></td><td>100000</td><td>100000</td><td>100000</td><td>100000</td></tr>
<tr><td>000000
000000
000000</td><td>000000
000000
000000</td><td>000000
000000
000000</td><td>000000
000000
000000</td><td>111111
000000
111111</td><td>111111
000000
000000</td><td>000000
111111
111111</td><td>000000
111111
000000</td></tr>
</table>

Figure 3: Structure of Generator Matrix for $b = 3$

row_i(right), then by adding a linear combination of the last 3 rows, we obtain a vector of weight 22, which is not doubly-even. Therefore we must place both ones in the same column-block.

There are $\begin{pmatrix} 6 \\ 2 \end{pmatrix} = 15$ ways to place 2 ones in a column-block, and thus 15 codewords of weight 2 in $RM^{\perp}$ may be obtained from each column-block. Exactly 5 of these 15 codewords are linearly independent. All 15 codewords in the first column-block may be generated by $row_4(\text{right}), \ldots, row_8(\text{right})$. Note that $row_4(\text{right}) + \ldots + row_8(\text{right}) + row_{22}(\text{right})$ is a vector with exactly 6 ones, all in the second column-block. Therefore all 15 codewords in the second column-block may be generated by $row_9(\text{right}), \ldots, row_{12}(\text{right})$ together with $row_4(\text{right}) + \ldots + row_{12}(\text{right}) + row_{22}(\text{right})$. Similarly, all 15 codewords in the third column-block may be generated by $row_{13}(\text{right}), \ldots, row_{16}(\text{right})$ and $row_4(\text{right}) + \cdots + row_8(\text{right}) + row_{13}(\text{right}) + \cdots + row_{16}(\text{right}) + row_{24}(\text{right})$. Finally, all 15 codewords in the fourth column-block may be generated by $row_{17}(\text{right}), \ldots, row_{20}(\text{right})$ and $row_4(\text{right}) + \cdots + row_{12}(\text{right}) + row_{17}(\text{right}) + \cdots + row_{20}(\text{right}) + row_{23}(\text{right}) + row_{24}(\text{right})$. All of the required 60 codewords of weight 2 in $RM^{\perp}$ have thus been generated. Therefore $row_{21}(\text{right})$ cannot have weight 2, and so we choose the next most restrictive possibility, namely weight 4.

4.2 Configuration on the Left-Hand Side

Observe that we have divided the remaining incomplete rows of the generator matrix into 5 sections. There are 4 row-blocks:
$\{row_4, \ldots, row_8\}$,
$\{row_9, \ldots, row_{12}\}$,
$\{row_{13}, \ldots, row_{16}\}$,
$\{row_{17}, \ldots, row_{20}\}$,
and a single row, $\{row_{21}\}$.

Now consider $row_i(\text{left})$, $4 \leq i \leq 21$. Based on the configuration in the first three rows, we may divide $row_i(\text{left})$ into 4 column-blocks, each of length 6.

Since $wt(row_i(\text{right})) = 2$ for $4 \leq i \leq 20$, and the minimum weight of C_{48} is 12, then $wt(row_i(\text{left})) \geq 12 - 2 = 10$ for $4 \leq i \leq 20$. Furthermore, if $wt(row_i(\text{left})) > 14$ then $wt(row_i + row_1 + row_2) < 12$. Therefore $wt(row_i(\text{left})) = 10$ or 14. However if $wt(row_i(\text{left})) = 14$ then $wt(row_i(\text{left}) + row_1(\text{left}) + row_2(\text{left})) = 10$. Therefore we may assume that $wt(row_i(\text{left})) = 10$ for $4 \leq i \leq 20$.

Similarly, since $wt(row_{21}(\text{right}) = 4$, and the minimum weight of C_{48} is 12, then $wt(row_{21}(\text{left})) \geq 12 - 4 = 8$. Furthermore, if $wt(row_{21}(\text{left})) > 16$ then $wt(row_{21} + row_1 + row_2) < 12$. Therefore we may assume that $wt(row_{21}(\text{left})) = 8$, 12 or 16. Observe that if $wt(row_{21}(\text{left})) = 16$ then $wt((row_{21} + row_1 + row_2)(\text{left})) = 8$ and therefore we may consider these two cases to be equivalent. Therefore we only need to consider the cases when $wt(row_{21}(\text{left})) = 8$ or 12.

Now let b_j be the number of blocks in $row_i(\text{left})$ which have j ones. Clearly $0 \leq j \leq 6$.

First let us show that either all blocks of $row_i(\text{left})$ have an even number of ones, or all blocks of $row_i(\text{left})$ have an odd number of ones. Clearly it is not possible for

there to be an odd number of blocks each containing an odd number of ones, for if there were, then row_i would have an odd weight.

Therefore suppose that the blocks of row_i(left) contain e_1, e_2, o_1 and o_2 ones, where e_1 and e_2 are even, and o_1 and o_2 are odd. Then one may always match up one even block, together with one odd block, against one of the linear combinations of the first 3 rows, creating a vector which is not doubly-even. For example, suppose that the first block of row_i(left) contains e_1 ones, the second block contains e_2 ones, the third block contains o_1 ones, and the fourth block contains o_2 ones. The total weight of row_i is $e_1 + e_2 + o_1 + o_2 + 2$. Adding row_i and row_3, we obtain a vector of weight $(6-e_1)+e_2+(6-o_1)+o_2+2$. Consider the difference between the weight of row_i and the weight of row_i+row_3. Since e_1 is even, $e_1-(6-e_1) \equiv 2 \pmod 4$. Since o_1 is odd, $o_1-(6-o_1) \equiv 0 \pmod 4$. Therefore $(e_1+e_2+o_1+o_2)-((6-e_1)+e_2+(6-o_1)+o_2) \equiv 2 \pmod 4$. Therefore row_i and $row_i + row_3$ cannot both be doubly-even. This is the difference between the weight of row_i and the weight of $row_i + row_3$, and therefore either all blocks of row_i are odd, or all blocks of row_i are even.

Since we have 4 blocks, we have:

$$b_0 + b_1 + b_2 + b_3 + b_4 + b_5 + b_6 = 4.$$

4.2.1 Row Types when row_i(left) has Weight 10

In this case, we have $\sum_{j=0}^{6} j \cdot b_j = 10$. If $b_6 = 1$ or $b_5 > 0$, then the block containing the largest number of ones has 5 or 6 ones and the block which contains the next-largest number of ones has at least 2 ones. We can always match up a block of 5 or 6 ones, together with the block which contains the next-largest number of ones, against 2 blocks of 6 ones from a linear combination of the first 3 rows, creating a vector of low weight. For example, suppose the first block of row_i(left) contains 6 ones, and the second block contains x ones, $x > 0$. Then the last 2 blocks contain $10 - 6 - x$ ones. Adding row_i and row_1, we obtain a vector of weight $2+(6-6)+(6-x)+10-6-x = 10 - 2x < 10$. This is less than the minimum weight of the code.

In the case $b_4 = 2$, one may similarly match up these 2 blocks of 4 ones against 2 blocks of 6 ones from a linear combination of the first 3 rows, creating a vector of low weight.

Therefore we have $b_6 = 0$, $b_5 = 0$, and $b_4 \leq 1$. Furthermore, either all blocks have an odd number of ones, or all blocks have an even number of ones. Therefore the only possibilities for the case when row_i(left) has weight 10 are: $\{b_0 = 0, b_1 = 0, b_2 = 3, b_3 = 0, b_4 = 1, b_5 = 0, b_6 = 0\}$ and $\{b_0 = 0, b_1 = 1, b_2 = 0, b_3 = 3, b_4 = 0, b_5 = 0, b_6 = 0\}$.

4.2.2 Row Types when row_i(left) has Weight 12

In this case, we have $\sum_{j=0}^{6} j \cdot b_j = 12$. If $b_6 > 1$ or $b_5 > 1$ then we can always match up the blocks of 5 or 6 ones against 2 blocks of 6 ones from a linear combination of the first 3 rows, creating a vector of low weight. If $b_6 = 1$ or $b_5 = 1$ then the block containing the largest number of ones has 5 or 6 ones, and the block containing the next-largest number of ones has at least 2 ones. We can always match up a block of 5 or 6 ones, together with the block which contains the next-largest number of ones, against 2 blocks of 6 ones from a linear combination of the first 3 rows, creating a

vector v such that v(left) has weight 8. Therefore we can consider this to be part of the case in which row_i(left) has weight 8. Therefore if row_i(left) has weight 12, then $b_6 = 0$ and $b_5 = 0$.

If $b_4 > 0$ then since either all blocks contain an even number of ones, or all blocks contain an odd number of ones, there are at least 2 blocks of 4 ones. We can match up 2 of these blocks of 4 ones against 2 blocks of 6 ones from a linear combination of the first 3 rows, again creating a vector v such that v(left) has weight 8. Therefore if row_i(left) has weight 12, $b_4 = 0$.

Hence the only possibility when row_i(left) has weight 12 is $\{b_0 = 0, b_1 = 0, b_2 = 0, b_3 = 4, b_4 = 0, b_5 = 0, b_6 = 0\}$.

4.2.3 Row Types when row_i(left) has Weight 8

In this case, we have $\sum_{j=0}^{6} j \cdot b_j = 8$. If we have only two blocks containing ones, then we can match up these two non-zero blocks against two blocks of 6 ones from a linear combination of the first 3 rows, creating a vector of low weight. As explained earlier, we can also eliminate all possibilities in which some blocks have an even number of ones, and some blocks have an odd number of ones. Therefore when row_i(left) has weight 8, the only possibilities are $\{b_0 = 0, b_1 = 3, b_2 = 0, b_3 = 0, b_4 = 0, b_5 = 1, b_6 = 0\}$, $\{b_0 = 1, b_1 = 0, b_2 = 2, b_3 = 0, b_4 = 1, b_5 = 0, b_6 = 0\}$, $\{b_0 = 0, b_1 = 0, b_2 = 4, b_3 = 0, b_4 = 0, b_5 = 0, b_6 = 0\}$ and $\{b_0 = 0, b_1 = 2, b_2 = 0, b_3 = 2, b_4 = 0, b_5 = 0, b_6 = 0\}$.

We note that Christine Bachoc [2] has shown that one can use a weight 12 codeword for row_{21}, in which row_{21}(left) has weight 4 and row_{21}(right) has weight 8. In this case, row_{21}(right) has one column-block containing 5 ones, and 3 column-blocks each containing a single one.

5 Search Method

A search assuming either remaining case would consist of attempting to complete a generator matrix of the specified form; we would attempt to complete this generator matrix row by row, starting with row_{b+1} and ending with row_{24-b}, where b is the dimension of the subcode RM. The first b rows and last b rows of the generator matrix are already known. As we proceed, we shall monitor the search to determine its size. For each row in turn, we must look at:

- the number of candidates for the row, and
- the number of survivors for the row.

A candidate is a survivor if it is compatible with all previously completed rows. A candidate is compatible with all previous rows if:

- it is linearly-independent from all previous rows, and
- all linear combinations of the candidate and previous rows are themselves valid codewords.

Note that this simple test for compatibility is potentially twice as expensive as each new row is added, since there are twice as many linear combinations to examine.

6 Complete Search

While it is still manageable, we shall find all non-isomorphic survivors for the current row as we proceed row by row.

In each of the remaining cases, we know the types of vectors which can be used to complete each row. When $b = 3$ we know that for row 4, either there is 1 block of 4 ones and 3 blocks of 2 ones, or there are 3 blocks of 3 ones and 1 block of a single one. When $b = 2$ we know that every remaining row has 1 block of 4 ones and 1 block of 6 ones.

Our first step, therefore, is to create a list of candidates, consisting of all vectors of the appropriate types. Note that each of these vectors is of length 24; as we consider each row, we must concatenate a length-24 vector with the appropriate "right-hand side" for that row as shown in either Fig. 2 or Fig. 3. This initial list is the list of candidates for row_{b+1}. We examine each candidate in turn to determine if it is compatible with all previous rows, and if so we add it to the list of survivors for row_{b+1}. We then check for isomorphisms among the subcodes generated by the completed rows. Since isomorphic codes have the same properties, we are only interested in non-isomorphic codes. Since the same set of types of vectors must be used to fill both row_{b+1} and row_{b+2}, and since row_{b+2} must be compatible with all of the rows with which row_{b+1} is compatible, our list of candidates for row_{b+2} consists ofall survivors from row_{b+1}, with of course the appropriate right-hand side row_{b+2}(right). Our list of survivors for row_{b+2} consists of all those candidates which are also compatible with row_{b+1}. For the same reasons, the above is true for each subsequent row up to the end of the first row-block. Namely, the list of candidates for row_i is the list of survivors for row_{i-1}, and the list of survivors for row_i is all those candidates which are also compatible with row_{i-1}. We shall now consider a method to find all possible non-isomorphic completions of a set of rows of the generator matrix. The rows in question must be consecutive rows within the same row-block. The algorithm we employ is as follows:

CompleteSearch(*StartRow*, *EndRow*)
for i = *StartRow* to *EndRow* *do*
begin
    for each possible non-isomorphic completion to row_{i-1} *do*
    begin
        for each vector v(left) in the list of survivors for row_{i-1} *do*
        begin
            concatenate v(left) with row_i(right), creating v;
            if v is compatible with $row_1, \ldots, row_{i-1}$ and $row_{25-b}, \ldots, row_{24}$
            then add v to the list of survivors for row_i;
            if the code C generated by v, $row_1, \ldots, row_{i-1}$ and $row_{25-b}, \ldots, row_{24}$ is
                non-isomorphic to all previously seen codes
            then C is a non-isomorphic completion to row_i;
        end
    end
end

Row	Non-Isomorphic Codes
4	2
5	4
6	35
7	877
8	52 161

Table 1: Number of Non-Isomorphic Codes by Row for $b = 3$

Row	Non-Isomorphic Codes
3	1
4	3
5	17
6	269
7	20 865

Table 2: Number of Non-Isomorphic Codes by Row for $b = 2$

The results of employing the above method, for the case $b = 3$, are shown in Table 1.

The results of employing the same method, for the case $b = 2$, are shown in Table 2.

6.1 Completing the Remainder of the Matrix

To complete the remainder of the matrix, we use the method described earlier in this paper, except that we do not do full isomorphism testing as it is now too expensive. To speed up the search, we used several specialized techniques, including those described in [9].

One of the specialized methods relies on the fact that the search is partitioned into three cases, namely, $b = 2, 3,$ or 4, based on the existence of a $(24, b, 12)$ subcode. Furthermore, a search assuming the case $b = 4$ was done previously. Thus, we assume that any codes arising from the case $b = 3$ do not contain a $(24, 4, 12)$ subcode. Similarly, we assume that any codes arising from the case $b = 2$ do not contain a $(24, 3, 12)$ subcode.

6.2 Results for $b = 3$

A few of the 52 161 codes complete to row_8 already contain a $(24, 4, 12)$ subcode. There remain 51 364 codes complete to row_8 which do not contain such a subcode.

However, testing for the existence of a subcode is expensive, and is not very effective for the case $b = 3$, especially in the early stages. So the next time we test for it is at the end of the search.

Recall that row_{21} has a different structure than rows 4 to 20. Hence we first exhaustively complete all the other rows, giving 23 rows in total. We found 10 524 generator matrices. Next, we test for the existence of a $(24, 4, 12)$ subcode. In fact, all

10 524 possbilities contain such subcodes. Therefore no $(48, 24, 12)$ self-dual doubly-even codes arise from the case $b = 3$.

The search assuming this case was completed on a network consisting of SGI R10000 and R12000 processors. We used autoson [16], a tool developed by Brendan D. McKay for scheduling processes across a network of workstations. The search took a total of approximately 4.7 CPU-years to complete.

6.3 Results for $b = 2$

There are 20 865 non-isomorphic codes complete to row 7. Again, we complete the generator matrix row-by-row. Since $b = 2$, we can reject any code which contains a $(24, 3, 12)$ subcode. This test turns out to be very effective for $b = 2$. At row 8, 20% of the codes are rejected. At row 9, 50% are rejected. At row 10, 90% are rejected, and at row 12, all codes are rejected. Therefore no $(48, 24, 12)$ self-dual doubly-even codes arise from the case $b = 2$.

The computation for this case took approximately 0.3 years of CPU time on a network of mixed SUN workstations. Again, we used autoson [16] for scheduling jobs over this network.

7 Conclusions

This completes the exhaustive search for $(48, 24, 12)$ self-dual doubly-even codes begun in [8, 10]. In the above works it is shown that the search for such codes may be divided into three cases, namely $b = 2, 3$ or 4, furthermore showing that the only code arising from the case $b = 4$ is the Extended Quadratic Residue Code of length 48.

We have shown that no codes arise from either the case $b = 3$ or the case $b = 2$. Thus we conclude that the Extended Quadratic Residue Code is the unique $(48, 24, 12)$ self-dual doubly-even code.

References

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

[2] Bachoc, C., private communication.

[3] Berlekamp, E. R., *Algebraic Coding Theory*, McGraw-Hill, 1968.

[4] Conway, J. H. and Pless, V., 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 Mathematics* 38 (1982), 143–156.

[5] Conway, J. H. and Pless, V., On the Enumeration of Self-Dual Codes, *J. Combin. Theory* 28 (1980), 26–53.

[6] Conway, J. H., Pless, V. and Sloane, N. J. A., The Binary Self-Dual Codes of Length up to 32: A Revised Enumeration, *J. Combin. Theory* 60 (1992), 183–195.

[7] Conway, J. H. and Sloane, N. J. A., A New Upper Bound on the Minimal Distance of Self-Dual Codes, *IEEE Trans. Inform. Theory* 36 (1990), 1319–1332.

[8] Houghten, S.K., "Construction of Extremal $(48, 24, 12)$ Doubly-Even Codes", Master's Thesis, Concordia University, 1993.

[9] Houghten, S.K., "On Combinatorial Searches for Designs and Codes", Doctoral Thesis, Concordia University, 1999.

[10] Houghten, S.K., Lam, C. and Thiel, L., Construction of $(48, 24, 12)$ Doubly-Even Self-Dual Codes, *Congr. Numer.* 103 (1994), 41–53.

[11] Huffman, W. C., Automorphisms of Codes with Applications to Extremal Doubly Even Codes of Length 48, *IEEE Trans. Inform. Theory* 28 (1982), 511–521.

[12] Huffman, W. C. and Yorgov, V. Y., A $(72, 36, 16)$ Doubly-Even Code Does Not Have an Automorphism of Order 11, *IEEE Trans. Inform. Theory* 33 (1987), 749–752.

[13] Koch, H.V., Unimodular Lattices and Self-dual Codes, Proceedings of the International Congress of Mathematicians, 1986.

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

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

[16] McKay, B.D., autoson — a distributed batch system for UNIX workstation networks (version 1.3), *Technical Report TR-CS-96-03*, Joint Computer Science Technical Report Series, Australian National University.

[17] Oral, H. and Phelps, K. T., Almost all Self-Dual Codes are Rigid, *J. Combin. Theory* 60 (1992), 264–276.

[18] Pless, V., 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.

[19] Pless, V., "Introduction to the Theory of Error-Correcting Codes", John Wiley and Sons, 1989.

[20] Pless, V. and Thompson, J. G., 17 Does Not Divide the Order of the Group of a $(72, 36, 16)$ Doubly-Even Code, *IEEE Trans. Inform. Theory* 28 (1982), 537–541.

[21] Schaathun, H. G., Private Communication, 2002.

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