Codes of length 2 correcting single errors of limited size

Torleiv Kløve

University of Bergen, N-5020 Bergen, Norway,
Torleiv.Klove@ii.uib.no

Abstract. Linear codes over $\mathbb{Z}_q$ of length 2, correcting single errors of size at most k, are considered. It is determined for which q such codes exists and explicit code constructions are given for those q. One case remains open, namely $q = (k+1)(k+2)$, where $k+1$ is a prime power. For this case we conjecture that no such codes exist.

Keywords: Error correcting codes, single errors, limited size errors

1 Introduction

Flash memories are non-volatile, high density and low cost memories. Flash memories find wide applications in cell phones, digital cameras, embedded systems, etc. and it is a major type of Non-Volatile Memory (NVM).

In order to improve the density of flash memories, multi- level (q-level) memory cells are used so that each cell stores $\log_2 q$ bits. Even though multi-level cells increase the storage density compared to single-level cells, they also impose two important challenges. The first one is that the voltage difference between the states is narrowed since the maximum voltage is limited. A natural consequence is that reliability issues such as low data retention and read/write disturbs become more significant. The errors in such cases are typically of limited magnitude.

The second major challenge in flash memory systems is that the writing mechanism is relatively very time consuming. A cell can be programmed from a lower level to a higher level by injecting additional amount of electrons in the floating gate. However, in order to program a cell from a higher level to lower level, an entire block of cells needs to be erased to zero and then using many iterations electrons are carefully injected to the floating gates of each and every cell to achieve the desired levels. Thus, rewriting a cell from the higher voltage level to a lower voltage level is quite expensive. The amount of time required for write operation can be reduced by using error correcting codes. The overshoot of voltage level while writing can be considered as asymmetric error of limited magnitude. Using codes capable of correcting limited magnitude asymmetric errors, the overshoot errors can be corrected. Because of this we do not need to be very precise about achieving the desired voltage level and so, the number of iterations required for charging the floating gates can be reduced, which in turn will reduce the write operations time.

2 Notations

We denote the set of integers by $\mathbb{Z}$. For $a, b \in \mathbb{Z}$, $a \leq b$, we let

$$[a, b] = \{a, a+1, a+2, \ldots, b\}.$$

For integers $q > 0$ and a, we let $(a \bmod q)$ denotes the main residue of a modulo q, that is, the least non-negative integer r such that q divides $a - r$.

Assume q is given. We denote modular addition of two integers a, b by $a \oplus b$, that is, $a \oplus b = ((a+b) \bmod q)$. Similarly, we define modular subtraction by $a \ominus b = ((a-b) \bmod q)$ and modular multiplication by $a \otimes b = (ab \bmod q)$.

We define the channel more precisely. Let q and k be integers, where $1 \leq k < q$. The alphabet is $\mathbb{Z}_q = [0, q-1]$. A symbol a in the alphabet $\mathbb{Z}_q$ may be modified during transmission into another symbol $a \oplus e \in \mathbb{Z}_q$ where e is an integer such that $|e| \leq k$. Error correcting codes for this channel have been considered in e.g. in [1], [5], [6]. Most of these are linear and single error correcting. The simplest non-trivial case are codes of length two. We consider this case in detail in this note.

3 General description of the codes

We define the codes and the problem precisely. Let $(a, b) \in \mathbb{Z}^2$. The corresponding code is

$$C = C_{a,b} = \{(u, v) \in \mathbb{Z}_q^2 \mid (a \otimes u) \oplus (b \otimes v) = 0\}.$$

When (u, v) is transmitted and (u', v') is received, the corresponding *syndrom* is $(a \otimes u') \oplus (b \otimes v')$. We see that if $(u', v') = (u \oplus e, v)$, the syndrom is

$$(a \otimes (u \oplus e)) \oplus (b \otimes v) = (a \otimes u) \oplus (a \otimes e) \oplus (b \otimes v) = a \otimes e.$$

Similarly, if $(u', v') = (u, v \oplus e)$, the syndrom is $b \otimes e$. Therefore, the code can correct a single error of size at most k if and only if the $1 + 4k$ syndroms

$$\{0\} \cup \{a \otimes e \mid e \in [-k, -1] \cup [1, k]\} \cup \{b \otimes e \mid e \in [-k, -1] \cup [1, k]\} \qquad (1)$$

are all distinct. If this is the case, we say that (a, b) is a (q, k) *check pair* or just a check pair if the values of q and k are clear from the context.

Our problem can now be precisely formulated as follows:

> For which q and k does a (q, k) check pair exist?

At first glance, this may seem to be a rather trivial problem. However, this appears not to be the case for all q and k. When a check pair exists, we also want describe the corresponding code and its encoding and decoding.

The following reformulation will be usefull.

Proposition 1 *For given q, k, $(a, b) \in \mathbb{Z}^2$ is a check pair if and only if all the following conditions are satisfied:*

1. $a \otimes e \neq b \otimes \varepsilon$ *for* $e, \varepsilon \in [-k,-1] \cup [1,k]$,
2. $\gcd(a,q) < q/(2k)$,
3. $\gcd(b,q) < q/(2k)$.

Proof. By the definitions, (a,b) is a check pair if and only if all the syndroms are distinct, that is, all the following conditions are satisfied:

1. $a \otimes e \neq b \otimes \varepsilon$ for $e, \varepsilon \in [-k,-1] \cup [1,k]$,
2. $a \otimes e \neq a \otimes \varepsilon$ for $-k \leq \varepsilon < e \leq k$,
3. $b \otimes e \neq b \otimes \varepsilon$ for $-k \leq \varepsilon < e \leq k$.

We will show that second of these conditions is equivalent to the second condition of the proposition and similarly for the third conditions. Let $d = \gcd(a,q)$. Then $\gcd(a/d, q/d) = 1$. Putting $z = e - \varepsilon$ we get the following chain of equivalent conditions:

$$\begin{aligned} 2) &\Leftrightarrow a \otimes z \not\equiv 0 \pmod{q} \text{ for all } z \in [1,2k] \\ &\Leftrightarrow (a/d) \otimes z \not\equiv 0 \pmod{(q/d)} \text{ for all } z \in [1,2k] \\ &\Leftrightarrow z \not\equiv 0 \pmod{(q/d)} \text{ for all } z \in [1,2k] \\ &\Leftrightarrow 2k < q/d \\ &\Leftrightarrow d < q/(2k). \end{aligned}$$

Similarly for the third condition.

Lemma 1. *Let (a,b) be a (q,k) check pair. Then*

1. *(b,a) is a check pair.*
2. *$(a,-b)$, $(-a,b)$, and $(-a,-b)$ are check pairs.*
3. *If $z \in \mathbb{Z}$ such that gcd$(q,z) = 1$, then (za, zb) is a check pair.*

Proof. The syndroms of (b,a) are clearly the same as the syndroms of (a,b). This proves case *1.* Also for case *2* the syndroms are the same.

Now, consider case *3.* Let $z' \otimes z = 1$. Multiplying by z', we see that

$$(za) \otimes e = (zb) \otimes \varepsilon \text{ if and only if } a \otimes e = b \otimes \varepsilon$$

for $e, \varepsilon \in [-k,-1] \cup [1,k]$. Further, $\gcd(za,q) = \gcd(a,q)$ and so

$$\gcd(za,q) < q/(2k) \text{ if and only if } \gcd(a,q) < q/(2k).$$

4 The case $q \leq (k+1)^2$

In [5], the following result was shown.

Theorem 1. *If $k \geq 1$ and $q \leq (k+1)^2$, then there are no (q,k) check pairs.*

It was also shown that $(1, k+1)$ *is* a $((k+1)^2+1, k)$ check pair. In this paper, we consider all $q > (k+1)^2$. We split the presentation into two parts:

The case $q \geq (k+1)^2+1$, $q \neq (k+1)(k+2)$. For this case we show in Section 5 that there exists a simple check pair.

The case $q = (k+1)(k+2)$. This is the hardest case. A check pair exists for some k, but not all. We discuss this case in Section 6.

5 The case $q \geq (k+1)^2 + 1$, $q \neq (k+1)(k+2)$

5.1 Check pairs

We will give explicit check pairs for all q in this case.

First, consider the pair $(1, k+1)$. The corresponding syndrom set is

$$[0,k] \cup [q-k, q-1] \cup \{(k+1)e \mid e \in [1,k]\} \cup \{q-(k+1)x \mid x \in [1,k]\}.$$

If $q - k(k+1) > k(k+1)$, that is, $q \geq 2k(k+1)+1$, then clearly all the syndroms are distinct and so $(1, k+1)$ is a check pair.

Similarly, if $q \in [(k+1)^2+1, 2k(k+1)-1]$ but $q \not\equiv 0 \pmod{k+1}$, then again all the syndroms are distinct.

It remains to consider $q \in \{x(k+1) \mid x \in [k+3, 2k]\}$. For these q we have $q \not\equiv 0 \bmod (k+2)$. By an argument similar to the one above, we see that that $(1, k+2)$ is a check pair.

We summarize these results in a theorem.

Theorem 2. *We have the following cases.*

1. *If $q \geq 2k(k+1)+1$, then $(1, k+1)$ is a check pair.*
2. *If $q \in [(k+1)^2+1, 2k(k+1)-1]$ but $q \not\equiv 0 \pmod{k+1}$, then $(1, k+1)$ is a check pair.*
3. *If $q \in \{x(k+1) \mid x \in [k+3, 2k]\}$, then $(1, k+2)$ is a check pair.*

5.2 The corresponding codes

We take a closer look at the codes corresponding to check pairs in the second case. The other cases are very similar. The code is

$$\begin{aligned} C_{1,k+1} &= \{(u,v) \mid u, v \in \mathbb{Z}_q, u \oplus ((k+1) \otimes v) = 0\} \\ &= \{((-(k+1)) \otimes v, v) \mid v \in \mathbb{Z}_q\}. \end{aligned}$$

The most natural encoding for the information $m \in \mathbb{Z}_q$ is to encode it into $((-(k+1)) \otimes m, m))$. In particular, this gives a systematic encoding.

For decoding, we assume that (u', v') is received and that at most one of the elements are in error, and by an amount e of size at most k. From this we want to recover the sent information. We look at the possible syndroms.

- If there are no errors, the syndrom is 0.
- If $u' = u \oplus e$ where $e \in [1,k]$, then the syndrom is $s = e$. In this case the second part is error free and so $m = v' = v$.
- If $u' = u \oplus e$ where $e \in [-k, -1]$, then the syndrom is $s = q + e$. Also in this case $m = v' = v$.
- If $v' = v \oplus e$ where $e \in [1,k]$, then the syndrom is $s = (k+1)e$ and so $e = s/(k+1)$. In this case $m = v' \ominus e = v' \ominus s/(k+1)$.
- If $v' = v \oplus e$ where $e \in [-k, -1]$, then the syndrom is $s = q + (k+1)e$ and so $e = (s-q)/(k+1)$ and $m = v' \ominus (s-q)/(k+1)$.

This gives the following decoding algorithm:

- if $s \in [0, k]$ or $s \in [q-k, q-1]$, then $m = v'$,
- else if ($s \bmod (k+1) = 0$, then $m = v' \ominus s/(k+1)$,
- else if ($(s-q) \bmod (k+1) = 0$, then $m = v' \ominus (s-q)/(k+1)$.

This gives a correct answer for all errors of the type we consider. Of course, if other types of errors have occurred, the decoding algorithm will either give a wrong answer or no answer at all (when none of the conditions are satisfied).

For codes corresponding to the first and third cases in Theorem 2, we we get a similar decoding algorithm.

6 The case $q = (k+1)(k+2)$

This is the main case.

6.1 An existence result

Theorem 3. *Let $k \geq 1$ and $q = (k+1)(k+2)$. For each integer a, $1 \leq a \leq q$, we have*

$$\gcd(a, q) > k$$

or there exists integers $x \in [1, k]$ and $y \in [-k, -1] \cup [1, k]$ such that

$$y = a \otimes x. \tag{2}$$

Remark. We see that (2) is equivalent to

$$ax - tq = y \tag{3}$$

for some integer t. We note that this implies that $\gcd(a, q)$ divides y. In particular, it implies that $\gcd(a, q) \leq |y| \leq k$.

We will use Farey-sequences in the proof. For a discussion of Farey-sequences, see e.g. [2, pages 23ff]. The Farey-sequence F_k is the sequence of fractions t/n, where $0 \leq t \leq n \leq k$ and $\gcd(t, n) = 1$, listed in increasing order. The size of F_k is $1 + \Phi_k$, where

$$\Phi = \Phi_k = \sum_{r=1}^{k} \varphi(r).$$

We denote the elements of F_k by t_i/n_i, where $t_0/n_0 = 0/1$ and $t_\Phi/n_\Phi = 1/1$.

Example 1. F_6 is

$$\frac{0}{1}, \frac{1}{6}, \frac{1}{5}, \frac{1}{4}, \frac{1}{3}, \frac{2}{5}, \frac{1}{2}, \frac{3}{5}, \frac{2}{3}, \frac{3}{4}, \frac{4}{5}, \frac{5}{6}, \frac{1}{1}.$$

We see that the elements are symmetric around $1/2$, and this is clearly a general property: we have

$$n_{\Phi-i} = n_i \text{ and } t_{\Phi-i} = n_i - t_i, \text{ that is, } t_{\Phi-i}/n_{\Phi-i} = 1 - t_i/n_i.$$

for $0 \leq i \leq \Phi$.

The following lemma contains Theorems 28 and 30 in [2].

Lemma 2. *Let t_i/n_i and t_{i+1}/n_{i+1} be consecuetive elements in F_k. Then*

$$t_{i+1}n_i - t_i n_{i+1} = 1, \tag{4}$$

and

$$n_i + n_{i+1} \geq k+1. \tag{5}$$

Let

$$s_i = \frac{t_i}{n_i} + \frac{1}{n_i(n_i+n_{i+1})}.$$

Lemma 3.

$$s_i = \frac{t_i}{n_i} + \frac{1}{n_i(n_i+n_{i+1})} = \frac{t_{i+1}}{n_{i+1}} - \frac{1}{n_{i+1}(n_i+n_{i+1})}. \tag{6}$$

Proof.

$$\begin{aligned}
&\Big(\frac{t_{i+1}}{n_{i+1}} - \frac{1}{n_{i+1}(n_i+n_{i+1})}\Big) - \Big(\frac{t_i}{n_i} + \frac{1}{n_i(n_i+n_{i+1})}\Big) \\
&\quad = \Big(\frac{t_{i+1}}{n_{i+1}} - \frac{t_i}{n_i}\Big) - \Big(\frac{1}{n_{i+1}(n_i+n_{i+1})} + \frac{1}{n_i(n_i+n_{i+1})}\Big) \\
&\quad = \frac{t_{i+1}n_i - t_i n_{i+1}}{n_i n_{i+1}} - \frac{1}{n_i n_{i+1}} = 0.
\end{aligned}$$

It is easy to show that Theorem 3 is true for $k \leq 3$. Therefore, Theorem 3 is equivalent to the following lemma (note that (8) is equivalent to (3)).

Lemma 4. *Let $k \geq 4$ and $q = (k+1)(k+2)$. For each integer a, $1 \leq a \leq q$, such that $\gcd(a,q) \leq k$, there exists integers x, y, and t such that $1 \leq x \leq k$,*

$$1 \leq |y| \leq k, \tag{7}$$

and

$$\frac{a}{q} - \frac{t}{x} = \frac{y}{xq}. \tag{8}$$

We have

$$\frac{t_i}{n_i} \leq \frac{a}{q} < \frac{t_{i+1}}{n_{i+1}}$$

for some i. We split the proof into cases. We first consider the cases when

$$\frac{t_i}{n_i} \leq \frac{a}{q} \leq s_i.$$

Case I, $\frac{t_i}{n_i} = \frac{a}{q}$ or, equivalently, $n_i a = t_i q$. Since $\gcd(n_i, t_i) = 1$, n_i must divide q. Hence $a = t_i(q/n_i)$, and so

$$\gcd(a,q) \geq \frac{q}{n_i} \geq \frac{q}{k} = \frac{k^2+3k+1}{k} > k+3 > k.$$

Case II, $\frac{t_i}{n_i} < \frac{a}{q} \leq \frac{t_i}{n_i} + \frac{1}{n_i(n_i+n_{i+1})}$. Then

$$0 < n_i a - t_i q \leq \frac{q}{n_i + n_{i+1}}.$$

Subcase IIa, $n_i + n_{i+1} \geq k+3$. Then

$$n_i a - t_i q \leq \frac{k^2+3k+2}{k+3} = k + \frac{2}{k+3} < k+1$$

and so $0 < n_i a - t_i q \leq k$.

Subcase IIb, $n_i + n_{i+1} = k+2$. Then

$$n_i a - t_i q \leq \frac{q}{k+2} = k+1.$$

Suppose that

$$n_i a - t_i q = k+1. \tag{9}$$

Then

$$a = \frac{t_i(k+2)+1}{n_i}(k+1). \tag{10}$$

From (4) we get

$$1 = t_{i+1}n_i - t_i n_{i+1} = t_{i+1}n_i - t_i(k+2) + t_i n_i$$

and so $(t_i + t_{i+1})n_i = t_i(k+2)+1$. Hence $\gcd(t_i + t_{i+1}, k+2) = 1$. Further, combining with (10) we get

$$a = (t_i + t_{i+1})(k+1).$$

Hence, $\gcd(a,q) = k+1 > k$.

Subcase IIc, $n_i + n_{i+1} = k+1$ is similar. First, from (4) we get, in this case,

$$1 = t_{i+1}n_i - t_i n_{i+1} = t_{i+1}n_i - t_i(k+1) + t_i n_i$$

and so

$$(t_i + t_{i+1})n_i = t_i(k+1)+1. \tag{11}$$

Hence $\gcd(t_i + t_{i+1}, k+1) = 1$ and

$$\gcd(n_i, k+1) = 1. \tag{12}$$

Further

$$n_i a - t_i q \leq \frac{q}{k+1} = k+2.$$

Subcase IIc-1,

$$n_i a - t_i q = k+2. \tag{13}$$

Then, by (11) and (13),

$$a = \frac{t_i(k+1)+1}{n_i}(k+2) = (t_i + t_{i+1})(k+2). \tag{14}$$

Hence, $\gcd(a, q) = k + 2 > k$.

Subcase IIc-2,

$$n_i a - t_i q = k + 1. \tag{15}$$

In this case,

$$n_i a = (t_i(k + 2) + 1)(k + 1),$$

and so, by (12), $n_i | (t_i(k + 2) + 1)$. Further, by (11), $n_i | (t_i(k + 1) + 1)$. Hence

$$n_i | ((t_i(k + 2) + 1) - (t_i(k + 1) + 1)) = t_i.$$

Since $\gcd(n_i, t_i) = 1$ and $t_i < n_i$, this is only possible if $n_i = 1$ and $t_i = 0$. Therefore, by (15), we must have $a = k + 1$ and so $\gcd(a, q) = k + 1 > k$.

Finally, we note that the cases where $s_i < \frac{a}{q} < \frac{t_{i+1}}{n_{i+1}}$ are similar. This completes the proof of Lemma 4 and so of Theorem 3.

Theorem 4. *Let $q = (k + 1)/k + 2)$. The pair $(1, a)$ is not a (q, k) check pair for any a.*

Proof. Suppose that $(1, a)$ is a check pair. By Proposition 1,

$$\gcd(a, q) < \frac{(k + 1)(k + 2)}{2k} < k.$$

By Theorem 3, there exist $e, \varepsilon \in [-k, -1] \cup [1, k]$ such that $e = a \otimes \varepsilon$. Hence, the syndroms are not all distinct. This contradicts our assumption that $(1, a)$ is a check pair.

Lemma 5. *Let $q = (k + 1)(k + 2)$. If (a, b) is a (q, k) check pair, then*

$$\gcd(a, q) > 1 \text{ and } \gcd(b, q) > 1.$$

Proof. Suppose that $\gcd(a, q) = 1$. Let a' be defined by $a' \otimes a = 1$ and let $b' = a' \otimes b$. By Lemma 1 part 3, $(1, b')$ is a check pair. However this contradicts Theorem 4. Hence, $\gcd(a, q) > 1$. Similarly, $\gcd(b, q) > 1$.

In contrast to this lemma, we have the following lemma.

Lemma 6. *Let $q = (k + 1)(k + 2)$. If (a, b) is a (q, k) check pair, then*

$$gcd(a, b, q) = 1.$$

Proof. Suppose that $\gcd(a, b, q) = d > 1$. Then we see that $(a/d, b/d)$ is a $(q/d, k)$ check pair:

- If $(a/d) \otimes e \equiv (b/d) \otimes \varepsilon \pmod{q/d}$ where $e, \varepsilon \in [-k, -1] \cup [1, k]$, then $a \otimes e \equiv b \otimes \varepsilon \pmod{q}$, but this is not possible since (a, b) is a (q, k) check pair.
- We have

$$\gcd\Big(\frac{a}{d}, \frac{q}{d}\Big) = \frac{\gcd(a, q)}{d} < \frac{q/(2k)}{d} = \frac{q/d}{2k}.$$

- Similarly,
$$\gcd\Big(\frac{b}{d}, \frac{q}{d}\Big) < \frac{q/d}{2k}.$$

However,
$$\frac{q}{d} \le \frac{(k+1)(k+2)}{2} < (k+1)^2,$$
and so no $(q/d, k)$ check pair exists by Theorem 1, a contradiction.

6.2 Check pairs when $k+1$ is not a prime power

Theorem 5. *Let $q = (k+1)(k+2)$. If $k+1 = \sigma\rho$ where $\gcd(\sigma, \rho) = 1$, then $(\sigma, \rho(k+2-\sigma))$ is a (q, k) check pair.*

Proof. Suppose that $k+1 = \sigma\rho$ where $\gcd(\sigma, \rho) = 1$. Then
$$q = (k+1)(k+2) = \sigma\rho(k+2).$$

We break the proof up into three parts.

1. We have $\gcd(\sigma, q) \le \sigma = (k+1)/\rho < (k+2)/2 < q/(2k)$.
2. We have
$$\gcd(\rho(k+2-\sigma), q) = \rho \gcd(k+2-\sigma, \sigma(k+2)) = \rho\, d.$$
We will show that $d = 1$. Since $\sigma|(k+1)$, we have $\gcd(\sigma, k+2) = 1$. Hence,
$$\gcd(k+2-\sigma, \sigma) = \gcd(k+2, \sigma) = 1$$
and
$$\gcd(k+2-\sigma, k+2) = \gcd(-\sigma, k+2) = 1.$$
Therefore $d = 1$ and so
$$\gcd(\rho(k+2-\sigma), q) = \rho < (k+2)/2 < q/(2k).$$
3. Suppose that
$$\sigma e \equiv \rho(k+2-\sigma)\varepsilon \pmod{\sigma\rho(k+2)}, \tag{16}$$
where $e, \varepsilon \in [-k, -1] \cup [1, k]$. Without loss of generality, we can assume that $\varepsilon \in [1, k]$. From (16) we get $\sigma e \equiv 0 \pmod{\rho}$ and so $e \equiv 0 \pmod{\rho}$, that is $e = \rho e'$. Since $|e| \le k = \sigma\rho - 1$, we have $1 \le |e'| \le \sigma - 1$. Similarly, we get $\varepsilon = \sigma\varepsilon'$ where $1 \le \varepsilon' \le \rho - 1$. Substituting these in (16) we get
$$\sigma\rho e' \equiv \rho(k+2-\sigma)\sigma\varepsilon' \pmod{\sigma\rho(k+2)}$$
and so
$$e' \equiv (k+2-\sigma)\varepsilon' \ (\mathrm{mod}\ k+2).$$
This implies that
$$-e' \equiv \sigma\varepsilon' \ (\mathrm{mod}\ k+2). \tag{17}$$

However, since

$$(-e') \bmod (k+2) \in [1, \sigma - 1] \cup [k+3-\sigma, k+1]$$

and

$$\sigma \le \sigma\varepsilon' \le \sigma(\rho - 1) = k + 1 - \sigma,$$

(17) is not possible. Hence, (16) is not possible.

6.3 Corresponding codes

We look closer at the codes corresponding to the check pairs of Theorem 5 and their encoding and decoding. The code is

$$C = \{(u, v) \mid u, v \in [0, q-1], \sigma u \oplus \rho(k+2-\sigma)v = 0\}.$$

Lemma 7. *We have*

$$C = \{(\rho U, \sigma V) \mid U \in [0, \sigma(k+2)-1], V \in [0, \rho(k+2)-1], U+V \equiv\ 0 \pmod{k+2}\}.$$

Proof. Since $\sigma u + \rho(k+2-\sigma)v \equiv\ 0 \pmod{\sigma\rho(k+2)}$, we get $\sigma u \equiv\ 0 \pmod{\rho}$. Since $\gcd(\sigma, \rho) = 1$, this implies that $u \equiv\ 0 \pmod{\rho}$. Hence $u = \rho U$ where $U \in [0, \sigma(k+2)-1]$.

Since $k+2 = \sigma\rho+1$, we similarly get $\rho v \equiv\ 0 \pmod{\sigma}$ and so $v \equiv\ 0 \pmod{\sigma}$ and $v = \sigma V$ where $V \in [0, \rho(k+2)-1]$. Finally, $(\rho U, \sigma V) \in C$ if and only if

$$\sigma\rho U \oplus \rho(k+2-\sigma)\sigma V \equiv\ 0 \pmod{\sigma\rho(k+2)}$$

which is equivalent to

$$U + V \equiv\ 0 \pmod{k+2}. \tag{18}$$

Corollary 1. *We have $|C| = q$.*

Proof. Let $V \in [0, \rho(k+2)-1]$. By (18), we have $(\rho U, \sigma V) \in C$ if and only if $U \equiv (-V) \pmod{\sigma(k+2)}$. Hence,

$$U \equiv (-V + z(k+2)) \pmod{\sigma(k+2)}$$

for some $z \in [0, \sigma - 1]$. Hence for each value of V there are σ possible values of U. Therefore, $|C| = \sigma\rho(k+2) = q$.

Theorem 4 showed that no systematic code exists in this case. However, also for the code given above there is an efficient bijection between $\mathbb{Z}_q$ and C.

The encoding (that is, the mapping from $\mathbb{Z}_q$ to C) can be done as follows: any integer $m \in [0, q-1]$ can be represented as

$$m = \sigma\mu + \nu \text{ where } \mu \in [0, \rho(k+2)-1], \nu \in [0, \sigma - 1].$$

We encode m into $((\rho(-\mu + \nu(k+2)) \bmod q), \sigma\mu)$.

The information can easily be recovered from the representation $(\rho U, \sigma V)$. First, we let $\mu = V$. Then we know that

$$\rho(-\mu + \nu(k+2)) \equiv \rho U \pmod{\rho\sigma(k+2)},$$

and so

$$-\mu + \nu(k+2) \equiv U \pmod{\sigma(k+2)},$$

which in turn implies that $U + \mu \equiv 0 \pmod{(k+2)}$ and so

$$\nu = \Big(\frac{U+\mu}{k+2} \bmod \sigma\Big) \text{ and } m = \sigma V + \nu.$$

We next consider the correction of errors. A codeword is $(u, v) = (\rho U, \sigma V)$ where (18) is satisfied.

- If $u' = u + e$ where $e \in [0, k]$, then the syndrom is $s = \sigma e$ and so $e = s/\sigma$.
- If $u' = u + e$ where $e \in [-k, -1]$, then $s = q + \sigma e$ and so $e = (s - q)/\sigma$.
- If $v' = v + e$, where $e \in [-k, -1] \cup [1, k]$, then

  $$s \equiv \ \rho(k + 2 - \sigma)e \pmod{\rho\sigma(k+2)}$$

  and so ρ divides s and

  $$\frac{s}{\rho} \equiv \ (k + 2 - \sigma)e \pmod{\sigma(k+2)}.$$

  We see that $\gcd(k + 2 - \sigma, \sigma(k + 2)) = 1$. Hence

  $$e \equiv f \overset{\text{def}}{=} ((k + 2 - \sigma)^{-1}\frac{s}{\rho} \bmod \sigma(k+2)),$$

  where the inverse is modulo $\sigma(k+2)$. If $f \le k$, then $e = f$. If $f \ge \sigma(k+2)-k$, then $e = f - \sigma(k + 2)$.

From this, we get the following decoding algorithm.

- if $s \equiv 0 \pmod{\sigma}$ and $s/\sigma \in [0, k]$, then decode into $(u \ominus (s/\sigma), v)$
- else if $s \equiv 0 \pmod{\sigma}$ and $s/\sigma \in [\rho(k + 2) - k, \rho(k + 2) - 1]$, then decode into $(u \ominus ((s - q)/\sigma), v)$
- else if $s \equiv 0 \pmod{\rho}$, let

  $$f = ((k + 2 - \sigma)^{-1}\frac{s}{\rho} \bmod \sigma(k+2)),$$

  - if $f \le k$, then decode into $(u, v \ominus f)$,
  - else decode into $(u, (v \ominus (f - \sigma(k + 2)) \bmod q))$.

For $k \le 100$ and $q = (k + 1)(k + 2)$, a complete search has shown that there are no check pairs when $k + 1$ a prime power. Possibly this is the case for all k and we formulate this a conjecture.

Conjecture 1. If $k+1$ a prime power, then there are no $((k+1)(k+2), k)$ check pairs.

When $k+1 \leq 42$ is not a prime power, all the (q,k) check pairs are those given by Theorem 5, combined with Lemma 1. Possibly this is the case in general.

Conjecture 2. If $k+1$ a not prime power and $q = (k+1)(k+2)$, then all (q,k) check pairs are congruent $(c\sigma, c\rho(k+2-\sigma))$ or $(-c\sigma, c\rho(k+2-\sigma))$ modulo q, where $k+1 = \sigma\rho$, $\gcd(\sigma,\rho) = 1$, and $\gcd(c,q) = 1$.

7 Summary

In this paper we have considered linear codes of length two over the alphabet $\mathbb{Z}_q = \{0, 1, \ldots, q-1\}$, correcting single errros at size at most k. It was well known [5] that for $q \leq (k+1)^2$ no such codes exist. For $q = (k+1)^2+1$ a simple code construction is known.

In this paper, we have studied the cases when $q \geq (k+1)^2+1$. In Section 5, we considered $q \neq (k+1)(k+2)$. We show that a simple code construction exists in all cases. We describe codes and their encoding and decoding, both quite simple.

In section 6 we considered $q = (k+1)(k+2)$. If $k+1$ is not a prime power, then we have found a code construction and again describe the codes, their encoding and decoding. This is the main result in this paper. For $k+1$ a prime power, we conjecture that no codes exist.

References

1. Elarief, N., Bose, B.: Optimal, systematic, q-ary codes correcting all asymmetric and symmetric errors of limited magnitude. IEEE Trans. Information Theory **56**, 979–983 (2010)
2. Hardy, G.H., Wright, E.M.: An Introduction to the Theory of Numbers (Fourth Edition). Oxford University Press (1962)
3. Jiang, A., Mateescu, R., Schwartz, M., Bruck, J.: Rank modulation for flash memories, IEEE Trans. Information Theory **55**, 2659–2673 (2009)
4. Kløve, T., Elarief, N., Bose, B.: Systematic, single limited magnitude error correcting codes for Flash Memories, IEEE Trans. Information Theory **57**, 4477–4487 (2011)
5. Kløve, T., Luo, J., Yari, S.: Codes correcting single errors of limited magnitude. IEEE Trans. Information Theory **58** 2206–2219 (2012)
6. Schwartz, M.: Quasi-cross lattice tilings with applications to flash memory. IEEE Trans. Information Theory **58** 2397–2405 (2012)
7. Yari, S., Kløve, T., Bose, B.: Some linear codes correcting single errors of limited magnitude for flash memories, IEEE Trans. Information Theory **59**, 7278–7287 (2013)