# Integer Codes for Flash Memories

**Hristo Kostadinov and Nikolai L. Manev**

**Abstract** This paper demonstrates the flexibility of integer codes with regard to various type of applications. New constructions of integer codes correcting asymmetric type of errors are proposed in the paper and how to apply the constructed codes to flash memories is discussed.

## 1 Introduction

Nonvolatile memory is computer memory that maintains stored information without a power supply. For example, the now ancient punch card is a type of nonvolatile memory because, thought it requires power to punch, it does not require power to remain punched. With the rise of portable electronic devices like cell phones, mp3 players, digital cameras, and PDAs, nonvolatile memory is increasingly important. Flash memory is currently the dominant nonvolatile memory because it is cheap and, unlike punch cards and other more recent kinds of nonvolatile memory, can be electrically programmed and erased with relative ease.

A chip of flash memory contains an array of tens of thousands of cells, and we assume that each chip stores a bit string. Each cell on a chip of flash memory can be thought of as a container of electrons. In binary flash each cell has two states: if there are electrons in the container then the cell is in the state 1, and if there are no electrons in the container, the cell is in state 0. Until recently, binary flash was the only kind of flash available, but now a new kind of flash memory has been developed, multilevel flash, that many see as the future of flash memory. In a multilevel cell, it is possible to distinguish between several different ranges of charge, allowing for more than two states.

H. Kostadinov (✉)
IMI-BAS, Acad. G. Bonchev St., Bl.8, 1113 Sofia, Bulgaria
e-mail: hristo@math.bas.bg

N.L. Manev
USEA "Lyuben Karavelov" and Institute of Mathematics and Informatics,
IMI-BAS, Acad. G. Bonchev St., Bl.8, 1113 Sofia, Bulgaria
e-mail: nlmanev@math.bas.bg



K. Georgiev et al. (eds.), *Advanced Computing in Industrial Mathematics*, Studies in Computational Intelligence 728,
https://doi.org/10.1007/978-3-319-65530-7_9

To transition between the states, it is necessary to add and remove electrons to and from the container. While it is easy to add electrons (i.e. to increase the state of the cell), it is impossible to remove electrons (i.e. to decrease the state of the cell) without first emptying the electrons from all the containers in a large selection of the chip. This process, called reset operation, is slow and, after many repetitions, wears out the chip. In multilevel flash, there are two types of mistakes that can occur when programming a cell: errors in which too many electrons are added ("overshoots") and errors in which too few electrons are added ("undershoots"). Because of the difficulty of removing electrons, overshoots are much bigger problem than undershoots. To avoid overshoots, the level of a cell is increased over multiple iterations by carefully adding small number of electrons at a time.

Flash devices exhibit a multitude of complex error types and behaviors, but common to all flavors of flash storage is the inherent asymmetry between cell programming (charge replacement) and cell erasing (charge removal). This asymmetry causes significant error sources to change cell levels in one dominant direction. Moreover, many reported common flash error mechanisms induce errors whose magnitudes (the number of error changes) are small, and independent of the alphabet size, which may be significantly larger than the typical error magnitude. In addition to the (uncontrolled) errors that challenge flash memory design and operation, codes for asymmetric limited-magnitude errors can be used to speed-up the memory access by allowing less-precise programming schemes that introduce errors in a controlled way. While not a panacea for all flash issues, the potential error migration and performance boost by asymmetric limited-magnitude codes, justify their addition, alongside other coding innovations, to the menu of flash coding solutions.

The most well-studied model for error-correcting codes is the model for symmetric errors. According to this model, a symbol, taken from the code alphabet, is changed to another symbol from the same alphabet, and all such are equally likely. The popularity of this model stems from both its applicability to a broad set of applications, and from the powerful construction techniques that were found to address it. In addition to the symmetric model, many other models, variations and generalizations were studied, each motivated by a behavior of practical systems or applications.

The asymmetric limited-magnitude error correcting codes can be used to speed up the writing process to flash devices (memory write is referred to as programming in the flash literature). This is done by relaxing the programming accuracy requirements, and using the codes to correct the resulting programming errors. Since the flash programming mechanism is inherently probabilistic, the introduction of "intentional" programming errors in a controlled way can significantly reduce the average programming time and improve the write performance. Such an outcome would be highly desirable given the inferiority of flash devices in write performance compared to their read performance, and to the sequential write performance of the hard-disk devices.

Asymmetric limited-magnitude error-correcting codes were proposed in [1]. The codes, proposed in that paper, were for the special case of correcting all asymmetric limited-magnitude errors within the codeword. These codes turn out to be a special case of the general construction method provided by Cassuto et al. [2].

In 2011, Klove and Bose [3] proposed systematic codes that correct single limited-magnitude systematic asymmetric errors and achieve higher rate than the ones given in [2]. They also showed how their code construction can be slightly modified to gives codes correcting symmetric errors of limited magnitude. Later Klove et al. [4] extended their result and gave a necessary and sufficient condition for existing a code over $GF(p)$ correcting a single asymmetric error.

As it has been already mentioned, asymmetric errors in flash memories are very common. However, there are cases in which the possible error type includes both a symmetric and an asymmetric error. For example, let us have a flash memory with $n$ voltage levels and have to increase the voltage level of a cell with current level $t-1$ by one (which is an usual situation when programming a flash memory). In such a case the most common error observed is overcharging the cell (increasing the level with at least 2, or to charge it less than is needed, i.e. after charging the cell stays at level $t-1$. Hence, that kind of error is a combination of the symmetric error ($\pm 1$) and the asymmetric error $(2, 3, \ldots, n)$.

The next of the paper is organized as follows. The necessary notations and definitions which are used in this paper are given in Sect. 2.1. In Sect. 2.2 we briefly discuss some existing results. New construction of integer codes for flash memory are presented in Sect. 3. Conclusion remarks and some open problems are discussed in Sect. 4.

## 2 Preliminaries

### 2.1 Integer Codes

Asymmetric error correcting codes were introduced by Varshamov and Tenegolz [5] in the middle of 60s. In that work they also gave the definition of integer code. For many years these codes have been almost forgotten. The appearance of multilevel flash memories renewed the interest in codes correcting asymmetric errors.

Integer codes are codes defined over finite rings of integers. Han Vinck and Morita [6] investigated integer codes with a view to magnetic recording and frame synchronization. A class of integer codes correcting specific types of errors and their application to coded modulation has been proposed by Kostadinov et al. [7]. Because of their flexibility integer codes are very suitable for application in multilevel flash memory.

**Definition 1** Let $\mathbb{Z}_A$ be the ring of integers modulo $A$. An ***integer code*** of length $n$ with parity-check matrix $\mathbf{H} \in \mathbb{Z}_A^{m \times n}$, is referred to be a subset of $\mathbb{Z}_A^n$, defined by

$$\mathscr{C}(\mathbf{H}, \mathbf{d}) = \{\mathbf{c} \in \mathbb{Z}_A^n \mid \mathbf{c}\mathbf{H}^T = \mathbf{d} \mod A\}$$

where $\mathbf{d} \in \mathbb{Z}_A^m$.

If $d = 0$ the code is a linear $[n, n-m]$ code over $\mathbb{Z}_A$. Without loss of generality, we can assume $d = 0$ in this paper. We write $\mathscr{C}(\mathbf{H})$, or only $\mathscr{C}$ if there is no possibility for ambiguity.

In this paper we consider codes with $m = 1$ (one check symbol only). Then $\mathbf{H} = (h_1, h_2, \ldots, h_n)$, $0 \neq h_i \in \mathbb{Z}_A$ and

$$\mathscr{C}(\mathbf{H}) = \{\mathbf{c} \in \mathbb{Z}_A \mid \sum_{i=1}^{n} c_i h_i = 0 \mod A\}$$

Integer codes are designed to correct specific type of errors instead of correcting number of errors in a codeword as it is the case with conventional codes. Thus, we need the following definition.

**Definition 2** Let $l_j$ and $e_i$ be positive integers, $j = 1, \ldots, m,\ i = 1, \ldots, s$. The code $\mathscr{C}(\mathbf{H}, d)$ is said to be a ***single*** $(l_1, l_2, \ldots, l_m, \pm e_1, \pm e_2, \ldots, \pm e_s)$***-error correctable*** if it can correct any single error with value $l_j$ or $\pm e_i$.

Obviously, $\mathscr{C}(\mathbf{H}, d)$ is a single $(l_1, l_2, \ldots, l_m, \pm e_1, \pm e_2, \ldots, \pm e_s)$-error correct-able code if and only if the subsets $\{h_j l_1, h_j l_2, \ldots, h_j l_m, \pm h_j e_1, \pm h_j e_2, \ldots, \pm h_j e_s\} \subset \mathbb{Z}_A$, are pairwise disjoint and of the same cardinality $2s + l$, for any $j = 1, 2, \ldots, n$. Thus, we have

$$A \geq (2s + l)n + 1.$$

**Definition 3** A single $(l_1, l_2, \ldots, l_m, \pm e_1, \pm e_2, \ldots, \pm e_s)$-error correctable code $\mathscr{C}(\mathbf{H}, d)$ of block length $n$ is called ***perfect***, when $A = (2s + l)n + 1$.

In most of the cases perfect integer codes do not exist. We shall say that a single $(l_1, l_2, \ldots, l_m, \pm e_1, \pm e_2, \ldots, \pm e_s)$-error correctable integer code $\mathscr{C}(\mathbf{H}, d)$ of block length $n$ over $\mathbb{Z}_A$ is ***optimal*** if $A$ is the minimum value for which the code $\mathscr{C}(\mathbf{H}, d)$ exists.

*Remark* One side effect, however, is that part of the power of the integer codes is used to correct wrap-around errors (i.e. errors modulo $A$), which does not appear in the flash memories. More precisely, we assume that a codeword $c$ may be changed into $c+e \pmod A$. If $c+e < 0$ or $c+e \geq A$, these are ***wraparound errors***. However, such errors usually constitute a minor part of the correctable errors. We can estimate this effect by a heuristic argument and show that when $A$ is large compared to the maximum value of the set $\{l_i, \pm e_j\}$, where $i = 1, \ldots, m$ and $j = 1, \ldots, s$, the main power of the code can be used to correct errors in flash memories.

## 2.2 Several Proposed q-ary Codes

In [2] Cassuto et al. describe a general method of constructing $t$-asymmetric $\lambda$-limited-magnitude error correcting codes from codes correcting symmetric errors.

Recently Klove et al. in [3] and [4] have done thorough study of $t$-asymmetric $\lambda$-limited-magnitude error correcting codes over $\mathbb{Z}_A$. Their study is based on the fact that the discussed coding problems can be reformulated and solve as problems in number theory.

**Definition 4** An error vector $\mathbf{e} = (e_1, e_2, \dots, e_n)$ is called a ***t-asymmetric λ-limited-magnitude error*** if $\mathrm{wt}(\mathbf{e}) = |\{i : e_i \neq 0\}| \leq t$ and $0 \leq e_i \leq \lambda$, for all $i = 1, 2, \dots, n$. A code $\mathscr{C}$ is called a ***t-asymmetric λ-limited-magnitude error correcting code*** if it can correct all $t$-asymmetric $\lambda$-limited-magnitude errors.

Let $E(\lambda, n, t)$ denote the set of all possible $t$ asymmetric $\lambda$-limited-magnitude error vectors of length $n$ over $A$.

In the cited papers the notation $B_t[\lambda](A)$ is used, or just $B_t[\lambda]$ when $A$ is known from the context. Namely, $B_t[\lambda](A)$ is defined as a set $B_t[\lambda](A) = \{b_1, b_2, \dots, b_n\}$ such that the set

$$\mathbf{e}B_t[\lambda](A) = \left\{ e_1b_1 + e_2b_2 + \cdots + e_nb_n \mid \mathbf{e} \in E(\lambda, n, t)\right\}$$

consists of distinct elements of $\mathbb{Z}_A$, i.e., modulo $A$. In these papers classes of codes correcting $t = n$ and $t = n - 1$ asymmetric $\lambda$-limited-magnitude errors are proposed. But the most attention was paid to the case $t = 1$, i.e., the set $B_1[\lambda](A)$. The Hamming bound for such codes gives $A \geq 1 + \lambda n$.

Define $M_\lambda(A)$ to be the maximal size of a $B_1[\lambda](A)$ set. In [3] it has been shown that for odd values of $A$ we have

$$M_\lambda(A) = \frac{A-1}{2} - \frac{\omega_A}{2}$$

where $\omega_A$ is the number of the cyclotomic cosets of odd size. In [4] $M_2(A)$ and bounds for $M_3(A)$ and $M_4(A)$ are determined.

In [4] a perfect $B_1[\lambda](p)$ sets for a class of primes $p$ is described. Also some results about $B_1[\lambda](A)$, $\lambda = 3, 4$, are obtained. Unfortunately theoretical results gives good codes for very large values of $A$. Optimal for codes over reasonable large alphabets are found by computer search in the case $t = 2$ and $t = n - 2$ for small $n$.

## 3 New Constructions of Integer Codes Correcting Single Type of Errors

In this Section we propose two constructions of integer codes correcting single errors. The next theorem gives the exact form of the check matrix of an integer code correcting a single asymmetric 2-limited-magnitude error.

**Theorem 1** *A 1-asymmetric 2-limited-magnitude error correctable code $\mathscr{C}$ of length $n$ over $Z_A$ has the following parity-check matrix* $\mathbf{H}$

- $\mathbf{H} = (1, 3, 5, \ldots, n-1, n+3, n+5, \ldots, 2n+1)$*, where* $A = 2n+2$ *and n is even*
- $\mathbf{H} = (1, 3, 5, \ldots, n-2, n+4, n+6, \ldots, 2n+3)$*, where* $A = 2n+4$ *and n is odd*

*Remark* In the case when $n$ is even the code is "almost" perfect—the exceeding is 1.

*Proof* Here we are going to prove the case when $n$ is even and $A = 2n+2$. The proof when $n$ is odd is analogous.

To show that a code $C$ with parity-check matrix

$$\mathbf{H} = (1, 3, 5, \ldots, n-1, n+3, n+5, \ldots, 2n+1)$$

is 1-asymmetric 2-limited-magnitude error correctable it is enough to prove that all elements of $\mathbf{H}_1 = 2\mathbf{H} \pmod{2n+2}$ are distinct and $\mathbf{H} \cap \mathbf{H}_1 = \emptyset$. We have

$$2H = (2, 6, 10, \ldots, 2n-10, 2n-6, 2n-2, 2n+6, 2n+10, \ldots, 4n-2, 4n+2)$$

and

$$H_1 = (2, 6, 10, \ldots, 2n-6, 2n-2, 4, 8, \ldots, 2n-4, 2n).$$

It is not so difficult one to see that all the elements in $\mathbf{H}_1$ are distinct. Moreover, the elements of $\mathbf{H}_1$ are even, while the elements of $\mathbf{H}$ are odd. So we have $\mathbf{H} \cap \mathbf{H}_1 = \emptyset$. With that the proof is completed.

Let $P_o$ be the set of odd primes $p$ such that $\mathrm{ord}_p(2)$ is odd. And let $A = 2n+2$ and $p|(A-1)$ where $p \in P_o$. According to Theorem 2 [4], it does not exist a 1-asymmetric 2-limited-magnitude error correctable code of length $n$ over $Z_{A-1}$. So, we can construct a 1-asymmetric 2-limited-magnitude error correctable code of length $n$ over $Z_A$ using Theorem 1, which is quasi-perfect. In such a way, we improve the result given in [4] in case of the length of the code $n$ such that $p|(2n+1), p \in P_o$.

Now we shall investigate how to construct an integer code $\mathscr{C}(\mathbf{H})$ capable to correct a single error of type $(\pm 1, 2)$. Because the code will be single error correctable, its check matrix $\mathbf{H}$ has to consist of a single row.

First, let us consider the set of integers

$$B = B(m) = \{4^k l < m \mid k, l, m \in \mathbb{N},\ m \geq 6 \text{ is even, and } l \text{ is odd}\}.$$

Let us divide the set $B$ into two subsets—$B_0$ and $B_1$, where

$$B_0 = \{a \in B \mid \exists\, b \in B : 2a + b \equiv 0 \pmod{2m}\} \quad \text{and} \quad B_1 = B \setminus B_0. \quad (1)$$

*Remark* Since $0 < a, b < m$ then $2a + b \equiv 0 \pmod{2m}$ is equivalent to $2a + b = 2m$. Hence $2a = 2m - b \leq 2m - 4$, i.e., $a \leq m - 2$. On the other hand $2m - 2a = b < m$ gives $a > m/2$. Therefore,

$$\frac{m}{2} < a \leq m - 2.$$

But not all integers in the above interval belongs to $B_0$. It is not difficult to prove that for $m = 2^k$ we have $B_0 = \emptyset$.

*Example 1* Let $m = 82$. Following the definition of $B$, $B_0$ and $B_1$ we obtain

$$B = \{4, 12, 16, 20, 28, 36, 44, 48, 52, 60, 64, 68, 76, 80\}$$

$$B_0 = \{44, 48, 52, 60, 64, 68, 76, 80\}$$

and

$$B_1 = \{4, 12, 16, 20, 28, 36\}.$$

We have the following construction for a single $(\pm 1, 2)$ error correctable integer code.

**Theorem 2** *Let $m \geq 6$ is a given integer and $m$ is even. Let us consider the sets $B(m), B_0$ and $B_1$. The integer code $\mathscr{C}(\mathbf{H})$ over $Z_{2m}$ with the check matrix*

$$\mathbf{H} = (1, 3, 5, 7, \dots, m - 1 \mid B_1)$$

*is a single $(\pm 1, 2)$ error-correctable.*

*Proof* The integer code $\mathscr{C}(\mathbf{H})$ is a single $(\pm 1, 2)$ error-correctable if all its syndrome values are different. Hence, to prove the theorem will be enough to show that

$$\mathbf{H} \cap (-\mathbf{H}) \cap (2\mathbf{H}) = \emptyset, \tag{2}$$

where all the operation are taken into $Z_{2m}$.

For convenience, let us divide $\mathbf{H}$ into 2 subsets $A_1 = (1, 3, 5, 7, \dots, m - 1)$ and $B_1$. So, the Eq. (2) is equivalent to

$$A_1 \cap (-A_1) \cap (2A_1) \cap B_1 \cap (-B_1) \cap (2B_1) = \emptyset. \tag{3}$$

One can easily see that $-A = (m + 1, m + 3, m + 5, \cdots, 2m - 1)$, and $A_1 \cap (-A_1) \cap (2A_1) = \emptyset$. Moreover,

$$A_1 \cup (-A_1) = \{2n + 1 | n = 0, 1, 2, 3 \ \dots, m - 1\} \tag{4}$$

and

$$2A_1 = \{4n + 2 | n = 0, 1, 2, 3 \ \dots, m/2 - 1\} \tag{5}$$

On the other side, $2m$ is divisible by 4. Hence, all the elements of the sets $B_1, -B_1 = \{2m - b | b \in B_1\}$ and $2B_1$ are divisible by 4. So, using (4) and (5) we have

$$(A_1 \cup (-A_1) \cup (2A_1)) \cap (B_1 \cup (-B_1) \cup (2B_1)) = \emptyset. \tag{6}$$

The only thing that we have to show is that

$$B_1 \cup (-B_1) \cup (2B_1) = \emptyset. \tag{7}$$

It is obvious that $B_1 \cup (2B_1) = \emptyset$, because all the elements of $B_1$ are not divisible by 8, while all the elements of $2B_1$ are divisible by 8. We have that $B_1 \cup (-B_1) = \emptyset$, since $2m - b_i > b_j$, where $b_i, b_j \in B_1$.

To prove that $(-B_1) \cup (2B_1) = \emptyset$ we should show that $2a + b \neq 0 \pmod{2m}$, where $a, b \in B_1$. But that follows from (1) and the definition of the set $B_1$. Hence, using (6) and (7) we complete the proof of the theorem.

*Example 2* Let $m = 64$. For the sets $B$, $B_0$ and $B_1$ we have

$$B_m = \{4, 12, 16, 20, 28, 36, 44, 48, 52, 60\}, \qquad B_0 = \emptyset,$$

$$B_1 = \{4, 12, 16, 20, 28, 36, 44, 48, 52, 60\}.$$

So, the integer code $\mathscr{C}(\mathbf{H})$ over $Z_{128}$ with the check matrix

$$\mathbf{H} = (1, 3, 5, 7, \dots, 63, 4, 12, 16, 20, 28, 36, 44, 48, 52, 60)$$

is a single $(\pm 1, 2)$ error-correctable. The length of the code is 42 and it is optimal. We can say that the code is "almost" perfect, because the exceeding is only 1.

Let $a \in B_0,\ b \in B_1$ and $2a + b \equiv 0 \pmod{2m}$. It is easy to see that if we change the elements $b$ with $a$ in $\mathbf{H}$ the theorem still holds.

## 4 Conclusion

In this work we have presented two new constructions of single error correctable integer codes designed for an application in a flash memory. Moreover, we gave the exact form of the check matrix for those codes. For some parameters, the obtained codes are optimal. The decoding complexity is linear, regarding to the code length, and can be used a look-up table to decode them. All these advantages of integer codes makes them very suitable for their usage in the practice. One can see that we only consider the case of single error and small magnitude. Actually, it is very difficult to obtain theoretical results for multiple errors and higher magnitude. On that we will focus for our future research [8].

**Acknowledgements** This work was partially supported by the National Science Fund of Bulgaria under Grant DFNI-I02/8.

## References

1. Ahlswede, R., Aydinian, H., Khachatrian, L.: Undirectional error control codes and related combinatorial problems. In: Proceedings of the Eighth International Workshop on Algebraic and Combinatorial Coding Theory (ACCT0'2002), pp. 6–9 (2002)
2. Cassuto, Y., Schwartz, M., Bohossian, V., Burck, J.: Codes for asymmetric limited-magnitude errors with application to multi-level flash memories. IEEE Trans. Inf. Theory **56**(4):1582–1595 (2010)
3. Klove, T., Bose, B.: Sistematic, single limited magnitude error correcting codes for flash memories. IEEE Trans. Inf. Theory **57**(7):4477–4487 (2011)
4. Klove, T., Lou, J., Naydenova, I., Yari, S.: Some codes correcting asymmetric errors of limited magnitude. IEEE Trans. Inf. Theory **57**(11):7459–7472 (2011)
5. Varshamov, R., Tenegolz, G.: One asymmetrical error-correctable codes. Automatika i Telematika **26**(2):288–292 (1965)
6. Han Vinck, A.J., Morita, H.: Codes over the ring of integers modulo $m$. IEICE Trans. Fundam. **E81-A**(10):2013–2018 (1998)
7. Kostadinov, H., Morita, H., Manev, N.: Integer codes correcting single errors of specific types $(\pm e_1, \pm e_2, \ldots, \pm e_s)$. IEICE Trans. Fundam. **E86-A**:1843–1849 (2003)
8. Kostadinov, H., Manev, N.: On codes for flash memories. In: International Workshop on Algebraic and Combinatorical Coding Theory (ACCT), pp. 197–202. Pomorie, Bulgaria (2012)