Научная статья
УДК 519.725, 512.623, 004.312
https://doi.org/10.31854/1813-324X-2025-11-2-109-120
EDN:VKAHMV

Метод вычисления полевой свертки на основе разложения многозначного расширенного поля Галуа

Илья Владимирович Ульянов, lopi2.lll@mail.ru

Академия Федеральной службы охраны Российской Федерации,
Орел, 302020, Российская Федерация

Аннотация

Актуальность. *Одной из нерешенных проблем теории помехоустойчивого кодирования остается проблема построения декодеров длинных кодов с низкой вычислительной сложностью. С точки зрения алгебраической теории кодирования краеугольным камнем для этого является операция умножения двух многочленов a и b над полем $GF(q^k)$ по модулю третьего многочлена g. С возрастанием q и k применение методов вычисления полевой свертки на основе операций логарифмирования и антилогарифмирования становится малоэффективным ввиду задействования большого объема памяти для построения таблиц. Упрощенные реализации полевой свертки, использующие несимметричность сопровождающей матрицы, и аналитические (не табличные) методы логарифмирования и антилогарифмирования, использующие полиномы Жегалкина, разработаны только для q = 2. Умножители на основе регистров сдвига обладают значительно меньшим быстродействием при больших q и k.*

Целью исследования *является поиск вариантов снижения вычислительной сложности операции полевой свертки в многозначных расширенных полях Галуа при ее синтезе в логическом базисе «И»–«ИЛИ»–«НЕ».*

Методы. *Проведен анализ однотактных методов умножения элементов многозначного расширенного поля Галуа, заданных в векторном или полиномиальном виде для различных степенных базисов. Приведены примеры вычисления полевых сверток в многозначных полях Галуа различными методами. Изучена структура рассматриваемого типа полей.*

Решение. *Показано, что операции сложения и умножения в поле GF(q), синтезированные на элементах логического базиса «И»–«ИЛИ»–«НЕ», вносят основной вклад в сложность итоговой логической схемы. Выявлено, что использование свойства разложения поля $GF(q^k)$ на подмножества по степени примитивного элемента поля GF(q) позволяет сократить число операций умножения. Предложен метод полевой свертки на основе матричного метода и преобразования Ганкеля – Теплица, учитывающий структуру поля, что позволяет сократить общее число логических элементов и повысить быстродействие проектируемого схемотехнического решения, а именно уменьшить цену по Квайну и ранг схемы. Дана сравнительная оценка разработанного метода.*

Новизна: *впервые предложен метод полевой свертки двух векторов в поле $GF(q^k)$, один из которых представлен в индикаторном виде.*

Теоретическая значимость. *Предложен новый метод вычисления полевой свертки на основе разложения многозначного расширенного поля Галуа. Доказано сокращение общего числа логических операций.*

Практическая значимость. *Предложенное решение может быть использовано при синтезе кодирующих-декодирующих устройств многозначных (символьных) кодов на элементах двоичной логики.*

Ключевые слова: *помехоустойчивое кодирование, полевая свертка, поле Галуа, матричный метод, преобразования Гаккеля – Теплица, цена по Квайну*

Ссылка для цитирования: Ульянов И.В. Метод вычисления полевой свертки на основе разложения многозначного расширенного поля Галуа // Труды учебных заведений связи. 2025. Т. 11. № 2. С. 109–120. DOI:10.31854/1813-324X-2025-11-2-109-120. EDN:VKAHMV

Original article
https://doi.org/10.31854/1813-324X-2025-11-2-109-120
EDN:VKAHMV

Method for Calculating Field Convolution Based on the Decomposition of a Multi-Valued Extended Galois Field

Ilya V. Ulyanov, lopi2.lll@mail.ru

Academy of the Federal Security Service of the Russian Federation,
Orel, 302020, Russian Federation

Annotation

Relevance. *One of the unsolved problems of the theory of error-correcting coding is the problem of constructing decoders of long codes with low computational complexity. From the point of view of algebraic coding theory, the cornerstone for this is the operation of multiplying two polynomials a and b over the field GF(q^k) modulo the third polynomial g. As q and k increase, the use of methods for calculating the field convolution based on the logarithm and antilogarithm operations becomes ineffective due to the use of a large amount of memory for constructing tables. Simplified implementations of the field convolution using the asymmetry of the accompanying matrix and analytical (non-tabular) methods of logarithm and antilogarithm using Zhegalkin polynomials have been developed only for q = 2. Multipliers based on shift registers have a significantly lower speed for large q and k.*

The aim of the study *is to find options for reducing the computational complexity of the field convolution operation in multivalued extended Galois fields during its synthesis in the logical basis "AND"–"OR"–"NOT".*

Methods. *An analysis of single-cycle methods for multiplying elements of a multivalued extended Galois field specified in vector or polynomial form for various power bases is carried out. Examples of calculating field convolutions in multivalued Galois fields by various methods are given. The structure of the considered type of fields is studied.*

Results. *It is shown that addition and multiplication operations in the field GF(q) synthesized on the elements of the logical basis "AND"–"OR"–"NOT" make the main contribution to the complexity of the resulting logical circuit. It is revealed that using the property of decomposition of the field GF(q^k) into subsets by the power of the primitive element of the field GF(q) allows to reduce the number of multiplication operations. A field convolution method based on the matrix method and the Hankel – Toeplitz transform is proposed, taking into account the field structure, which allows to reduce the total number of logical elements and increase the performance of the designed circuit solution, namely, to reduce the Quine price and the circuit rank. A comparative assessment of the developed method is given.*

Scientific novelty. *For the first time, a method of field convolution of two vectors in the field GF(q^k) is proposed, one of which is presented in the indicator form.*

Theoretical / Practical significance. *A new method for calculating the field convolution based on the decomposition of a multivalued extended Galois field is proposed. Reduction of the total number of logical operations is proved. The proposed solution can be used in the synthesis of encoding and decoding devices for multi-valued (symbolic) codes on binary logic elements.*

Keywords: *error-correcting coding, field convolution, Galois field, matrix method, Gakkel – Toeplitz transforms, Quine price*

For citation: Ulyanov I.V. Method for Calculating Field Convolution Based on the Decomposition of a Multi-Valued Extended Galois Field. *Proceedings of Telecommunication Universities.* 2025;11(2):109–120. (in Russ.) DOI:10.31854/1813-324X-2025-11-2-109-120. EDN:VKAHMV

Введение

Одной из фундаментальных проблем теории кодирования, препятствующих построению длинных кодов, является трудоемкость операции умножения двух многочленов *a* и *b* над полем *GF*(q^k) по модулю третьего многочлена *g*. Такая

операция называется сверткой многочленов. Особый интерес с теоретической и практической точки зрения представляет полевая свертка, когда многочлен g неприводим над полем $GF(q^k)$. Полевая свертка в качестве бинарной операции умножения в совокупности с бинарной операцией сложения являются основой для формирования поля $GF(q^k)$, состоящего из множества всевозможных многочленов-остатков по модулю неприводимого многочлена g. Очевидно, что умножение самих элементов (многочленов-остатков) также выполняется с помощью полевой свертки [1–3]. Другие виды сверток, а именно линейная и циклическая, в работе не рассматриваются. Применимость предлагаемого метода для их реализации является предметом дальнейших исследований автора.

В теории и практике кодирования информации существует ряд подходов к вычислению полевой свертки. В общем случае аналитическое выражение для прямого вычисления полевой свертки достаточно громоздкое. Поэтому для синтеза аппаратной реализации на этапе проектирования по заданному многочлену-константе g итерационно просчитывают рекуррентную формулу и получают компактную прямую формулу, на основе которой разрабатывают устройство [1].

При малых q и k с точки зрения аппаратной реализации предпочтительным является синтез устройства на основе компактной прямой формулы. В случае же увеличения q и k возрастает вычислительная сложность такого подхода, поэтому находят свое применение таблицы логарифмов и антилогарифмов. Это кардинально снижает вычислительную сложность, однако требует значительных объемов памяти и вносит задержки на поиск в соответствующих таблицах. Дальнейшее увеличение q и k приводит к невозможности использования таблиц логарифмов и антилогарифмов в виду их значительного объема. Это требует разработки эффективных с вычислительной точки зрения и с точки зрения используемой памяти алгоритмов расчета полевой свертки, рассчитываемой за один такт для больших q и k. Стоит отметить, что существуют многотактные умножители произвольных элементов поля $GF(q^k)$, которые синтезируются на основе регистров сдвига [4, 5]. В данной работе они не рассматриваются по причине своей нерегулярности.

В исследованных автором источниках [1–21], кроме, пожалуй [2, 6], рассматриваются вопросы практической реализации операций в полях $GF(q^k)$, где $q = 2$. Предлагаемый далее метод предполагает $q > 2$, т. е. полевая свертка рассматривается в многозначных (символьных) полях Галуа. При $q = 2$ предлагаемый метод вырождается в общеизвестный матричный метод вычисления полевой свертки [2].

Стоит отметить, что по тексту статьи при указании поля Галуа в общем виде – $GF(q^k)$ подразумевается, что оно образовано соответствующим неприводимым полиномом $P(x)$. Если же поле Галуа указано с конкретными значениями q и k, например $GF(3^3)$, то это значит, что поле образовано полиномом, указанным в соответствующем примере, ссылка на который дается по тексту.

Логарифмирование–антилогарифмирование

Наиболее простым с вычислительной точки зрения является полевая свертка на основе операций логарифмирования–антилогарифмирования (рисунок 1). Под логарифмом в расширенном поле $GF(q^k)$ понимают степень i, в которую необходимо возвести примитивный элемент ε, чтобы получить рассматриваемый элемент поля ε^i: $i = \log_\varepsilon \varepsilon^i$ [1]. Множество логарифмов $\{0, 1, \ldots, q^k - 2\}$ поля $GF(q^k)$ с определенными на нем бинарными операциями сложения и умножения образует коммутативное кольцо по модулю $q^k - 1$.

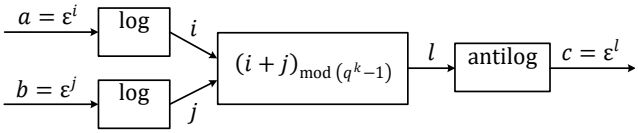

Рис. 1. Структурная схема умножения в поле $GF(q^k)$ на основе операций логарифмирования–антилогарифмирования

Fig. 1. Block Diagram of Multiplication in the Field $GF(q^k)$ Based on Logarithm–Antilogarithm Operations

Пример 1. Вычисление полевой свертки с помощью логарифмирования–антилогарифмирования

Рассмотрим свертку элементов поля $GF(3^3)$, построенного на основе неприводимого многочлена $P(x) = x^3 + 2x + 1$ (основание кода $q = 3$, порядок – $k = 3$). В таблице 1 приведены элементы поля для левого степенного базиса $(1, \varepsilon, \varepsilon^2)$:

ТАБЛИЦА 1. Элементы поля $GF(3^3)$ по основанию $P(x) = x^3 + 2x + 1$ для левого степенного базиса

TABLE 1. Elements of the Field $GF(3^3)$ to the Base $P(x) = x^3 + 2x + 1$ for the Left Power Basis

Десятичный номер	Степень ε^i	Логарифм i	Вектор
1	ε^0	0	(1 0 0)
2	ε^1	1	(0 1 0)
3	ε^2	2	(0 0 1)
4	ε^3	3	(2 1 0)
5	ε^4	4	(0 2 1)
6	ε^5	5	(2 1 2)
7	ε^6	6	(1 1 1)
8	ε^7	7	(2 2 1)
9	ε^8	8	(2 0 2)
10	ε^9	9	(1 1 0)
11	ε^{10}	10	(0 1 1)

Десятичный номер	Степень ε^i	Логарифм i	Вектор
12	ε^{11}	11	(2 1 1)
13	ε^{12}	12	(2 0 1)
14	ε^{13}	13	(2 0 0)
15	ε^{14}	14	(0 2 0)
16	ε^{15}	15	(0 0 2)
17	ε^{16}	16	(1 2 0)
18	ε^{17}	17	(0 1 2)
19	ε^{18}	18	(1 2 1)
20	ε^{19}	19	(2 2 2)
21	ε^{20}	20	(1 1 2)
22	ε^{21}	21	(1 0 1)
23	ε^{22}	22	(2 2 0)
24	ε^{23}	23	(0 2 2)
25	ε^{24}	24	(1 2 2)
26	ε^{25}	25	(1 0 2)

Множители:

$$a = (a_0 \quad a_1 \quad a_2) = \varepsilon^8 = (2 \quad 0 \quad 2),$$
$$b = (b_0 \quad b_1 \quad b_2) = \varepsilon^{13} = (2 \quad 0 \quad 0).$$

Результат:

$$c = (c_0 \quad c_1 \quad c_2) = a \times b = \varepsilon^8 \times \varepsilon^{13} = \\ = \varepsilon^{21} = (1 \quad 0 \quad 1).$$

Матричный способ

В ряде случаев вычислять полевую свертку удобнее на основе прямых выражений. Одним из способов получения таких выражений является матричный способ [2] (рисунок 2).

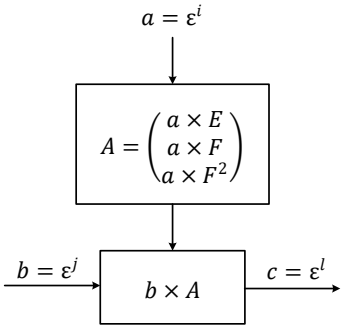

Рис. 2. Блок-схема матричного способа умножения элементов поля $GF(q^k)$

Fig. 2. Block Diagram of the Matrix Method for Multiplying Elements of the Field $GF(q^k)$

Пример 2. Вычисление полевой свертки матричным способом

Синтезируем прямые выражения для вычисления полевой свертки произвольных элементов поля, рассмотренного в примере 1:

1) определим сопровождающую матрицу и вычислим ее степени:

$$F = \begin{pmatrix} \varepsilon^1 \\ \varepsilon^2 \\ \varepsilon^3 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 2 & 1 & 0 \end{pmatrix}, F^2 = \begin{pmatrix} 0 & 0 & 1 \\ 2 & 1 & 0 \\ 0 & 2 & 1 \end{pmatrix};$$

2) вычислим составную матрицу:

$$A = \begin{pmatrix} a \times E \\ a \times F \\ a \times F^2 \end{pmatrix} = \begin{pmatrix} a_0 & a_1 & a_2 \\ 2a_2 & a_0 + a_2 & a_1 \\ 2a_1 & a_1 + 2a_2 & a_0 + a_2 \end{pmatrix},$$

где $E = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$;

3) синтезируем прямые выражения для вычисления результата $c = b \times A = (c_0 \quad c_1 \quad c_2)$:

$$c_0 = a_0 b_0 + 2a_1 b_2 + 2a_2 b_1, \quad (1)$$
$$c_1 = a_0 b_1 + a_1 b_0 + a_1 b_2 + a_2 b_1 + 2a_2 b_2, \quad (2)$$
$$c_2 = a_0 b_2 + a_1 b_1 + a_2 b_0 + a_2 b_2; \quad (3)$$

4) примеры вычисление сверток для произвольных элементов поля:

$$a = \varepsilon^8 = (2 \quad 0 \quad 2), b = \varepsilon^{13} = (2 \quad 0 \quad 0);$$
$$c_0 = 2 \cdot 2 + 2 \cdot 0 \cdot 0 + 2 \cdot 2 \cdot 0 = 1,$$
$$c_1 = 2 \cdot 0 + 0 \cdot 2 + 0 \cdot 0 + 2 \cdot 0 + 2 \cdot 2 \cdot 0 = 0,$$
$$c_2 = 2 \cdot 0 + 0 \cdot 0 + 2 \cdot 2 + 2 \cdot 0 = 1;$$
$$c = \varepsilon^{21} = (1 \quad 0 \quad 1);$$
$$a = \varepsilon^{11} = (2 \quad 1 \quad 1), b = \varepsilon^{20} = (1 \quad 1 \quad 2);$$
$$c_0 = 2 \cdot 1 + 2 \cdot 1 \cdot 2 + 2 \cdot 1 \cdot 1 = 2;$$
$$c_1 = 2 \cdot 1 + 1 \cdot 1 + 1 \cdot 2 + 1 \cdot 1 + 2 \cdot 1 \cdot 2 = 1;$$
$$c_2 = 2 \cdot 2 + 1 \cdot 1 + 1 \cdot 1 + 1 \cdot 2 = 2;$$
$$c = \varepsilon^5 = (2 \quad 1 \quad 2).$$

Преобразование Ганкеля – Теплица

Другим подходом, позволяющим при вычислениях над полями Галуа обходиться без операций логарифмирования и антилогарифмирования, является применения матриц Ганкеля (4) и Теплица (5) [3]. Выражение для полевой свертки элементов поля $GF(2^k)$, заданных в векторном или полиномиальном виде, может быть реализовано с помощью специальных векторно-матричных конструкций (рисунок 3). Элементами таких матриц являются модифицированные логарифмы поля $GF(2^k)$ (десятичные номера из таблиц логарифмирования-антилогарифмирования), которые затем заменяются на вектора соответствующих элементов поля.

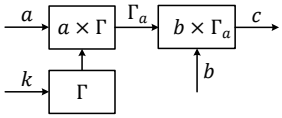

Рис. 3. Блок-схема умножения элементов поля $GF(2^k)$ на основе преобразования Ганкеля – Теплица

Fig. 3. A Block Diagram for Multiplying Elements of the $GF(2^k)$ Field Based on the Hankel – Toeplitz Transformation

$$\Gamma = \begin{pmatrix} 1 & 2 & \cdots & k \\ 2 & 3 & \cdots & (k+1) \\ \vdots & \vdots & \vdots & \vdots \\ k & (k+1) & \cdots & (2k-1) \end{pmatrix}, \quad (4)$$

$$\mathrm{T} = \begin{pmatrix} k & \cdots & 2 & 1 \\ (k+1) & \cdots & 3 & 2 \\ \vdots & \vdots & \vdots & \vdots \\ (2k-1) & \cdots & (k+1) & k \end{pmatrix}. \quad (5)$$

Очевидно, что матрицы Γ и T связаны с помощью операции перестановки столбцов: $\acute{\Gamma} = \mathrm{T}$.

При замене элементов матрицы Γ соответствующими векторами, например, для поля $GF(2^3)$ с $(x) = x^3 + x + 1$ и правым степенным базисом $(\varepsilon^2, \varepsilon, 1)$, получим следующее векторное представление матрицы Γ:

$$\Gamma = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 4 & 5 \end{pmatrix} = \begin{pmatrix} \varepsilon^0 & \varepsilon^1 & \varepsilon^2 \\ \varepsilon^1 & \varepsilon^2 & \varepsilon^3 \\ \varepsilon^2 & \varepsilon^3 & \varepsilon^4 \end{pmatrix} = \begin{pmatrix} (001) & (010) & (100) \\ (010) & (100) & (011) \\ (100) & (011) & (110) \end{pmatrix}.$$

Вычисление полевой свертки в полях $GF(2^k)$ производится в соответствии со схемой (см. рисунок 3), которая может быть записана в ΣΠ-форме следующим образом [3]:

$$c = \sum_{i=0}^{k-1} a_i \sum_{j=0}^{k-1} b_j \varepsilon^{i+j}, \quad (6)$$

где $c = (c_k \quad \cdots \quad c_1 \quad c_0)$ – вектор-результат; $a = (a_k \quad \cdots \quad a_1 \quad a_0)$ и $b = (b_k \quad \cdots \quad b_1 \quad b_0)$ – векторы-множители.

Пример 3. Вычисление полевой свертки с помощью преобразования Ганкеля – Теплица

Предложенный для полевой свертки в полях $GF(2^k)$ [3] подход может быть использован для полей $GF(q^k)$. Рассмотрим поле $GF(3^3)$, построенное в примере 1, с оговоркой для правого степенного базиса (векторное представление элементов поля будет взято с операцией перестановки столбцов по отношению к данным, указанным в таблице 1).

Раскроем вначале правую и затем левую суммы в выражении (6):

$$c = \sum_{i=0}^{2} a_i (b_0 \varepsilon^{i+0} + b_1 \varepsilon^{i+1} + b_2 \varepsilon^{i+2}) =$$
$$= a_0 b_0 \varepsilon^0 + a_0 b_1 \varepsilon^1 + a_0 b_2 \varepsilon^2 +$$
$$+ a_1 b_0 \varepsilon^1 + a_1 b_1 \varepsilon^2 + a_1 b_2 \varepsilon^3 + a_2 b_0 \varepsilon^2 + a_2 b_1 \varepsilon^3 +$$
$$+ a_2 b_2 \varepsilon^4 = a_0 b_0 (0 \quad 0 \quad 1) + a_0 b_1 (0 \quad 1 \quad 0) +$$
$$+ a_0 b_2 (1 \quad 0 \quad 0) + a_1 b_0 (0 \quad 1 \quad 0) + a_1 b_1 (1 \quad 0 \quad 0) +$$
$$+ a_1 b_2 (0 \quad 1 \quad 2) + a_2 b_0 (1 \quad 0 \quad 0) +$$
$$+ a_2 b_1 (0 \quad 1 \quad 2) + a_2 b_2 (1 \quad 2 \quad 0).$$

Таким образом, компоненты вектора c могут быть вычислены следующим образом:

$$c_2 = a_0 b_2 + a_1 b_1 + a_2 b_0 + a_2 b_2, \quad (7)$$

$$c_1 = a_0 b_1 + a_1 b_0 + a_1 b_2 + a_2 b_1 + 2 a_2 b_2, \quad (8)$$

$$c_0 = a_0 b_0 + 2 a_1 b_2 + 2 a_2 b_1. \quad (9)$$

Очевидно, что выражения (7–9) идентичны выражениям (1–3). Это следует из того, что по своей сути преобразование Ганкеля – Теплица (ГТ-преобразование) является произведением исходного вектора a (или его перестановки $\acute{a}$) на примитивный элемент поля $GF(q^k)$ α [3]:

$$a\Gamma = [(\acute{a})(\acute{a}\alpha) \ldots (\acute{a}\alpha^{k-1})]$$

и

$$\acute{a}\Gamma = [(a)(a\alpha) \ldots (a\alpha^{k-1})].$$

На рисунке 4 показана структурная схема полевой свертки произвольных элементов поля $GF(3^3)$, определенного в примере 1, демонстрирующая вычисления векторов, заданных в примере 2. Схема синтезирована на основе выражений (1–3). Сложение и умножение выполняются в поле $GF(3^3)$.

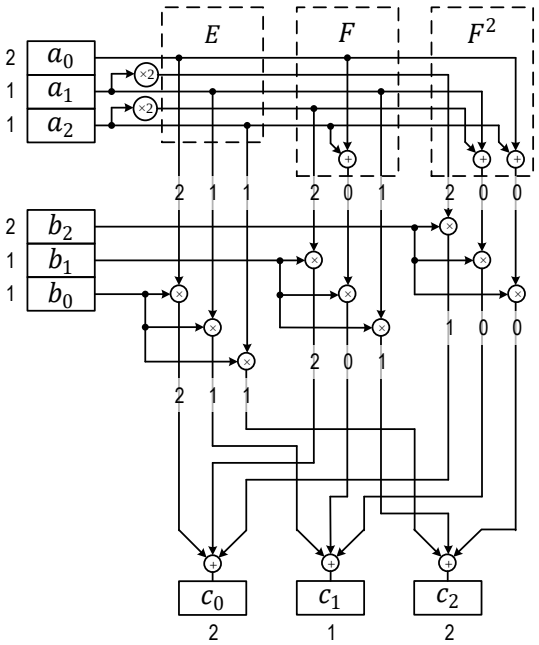

Рис. 4. Функциональная схема матричного метода умножения произвольных элементов поля $GF(3^3)$

Fig. 4. Functional Diagram for the Matrix Multiplication Method for Arbitrary Elements of the Field $GF(3^3)$

Структурная схема (см. рисунок 4) проста с теоретической точки зрения. Однако ее синтез на двоичных логических элементах в программируемых логических интегральных схемах (ПЛИС) или больших интегральных схемах (БИС) для больших q и k приводит к резкому росту цены (стоимости) по Квайну и снижению быстродействия (повышению ранга). Это связано с необходимостью синтеза сумматоров и умножителей в полях $GF(q^k)$ на ос-

нове соответствующих двоичных операций из поля $GF(2^k)$.

Пример 4. Синтез логических операций в поле $GF(3^3)$ на основе элементов логического базиса «И»–«ИЛИ»–«НЕ»

Синтезируем совершенные дизъюнктивные нормальные формы (СДНФ) для частичных булевых функций в поле $GF(3^3)$ из примера 1, реализующих соответствующие операции (см. рисунок 4). Для этого построим таблицу 2.

ТАБЛИЦА 2. Таблица истинности для логических операций в поле *GF*(3^3)

TABLE 2. Truth Table for Logical Operations in the Field GF(3^3)

a_i	a_i^1	a_i^0	b_i	b_i^1	b_i^0	$(a_i \oplus b_i)_{\mathrm{mod}3}$			$(a_i \otimes b_i)_{\mathrm{mod}3}$			$(a_i \otimes 2)_{\mathrm{mod}3}$		
						c_i	c_i^1	c_i^0	c_i	c_i^1	c_i^0	c_i	c_i^1	c_i^0
0	0	0	**0**	0	0	**0**	0	0	**0**	0	0	**0**	0	0
1	0	1	**0**	0	0	**1**	0	1	**0**	0	0	**2**	1	0
2	1	0	**0**	0	0	**2**	1	0	**0**	0	0	**1**	0	1
0	0	0	**1**	0	1	**1**	0	1	**0**	0	0	–	–	–
1	0	1	**1**	0	1	**2**	1	0	**1**	0	1	–	–	–
2	1	0	**1**	0	1	**0**	0	0	**2**	1	0	–	–	–
0	0	0	**2**	1	0	**2**	1	0	**0**	0	0	–	–	–
1	0	1	**2**	1	0	**0**	0	0	**2**	1	0	–	–	–
2	1	0	**2**	1	0	**1**	0	1	**1**	0	1	–	–	–

Получены следующие СДНФ:

– для операции $(a_i \oplus b_i)_{\mathrm{mod}3}$:

$$c_i^1 = a_i^1\overline{a_i^0}\,\overline{b_i^1}\,\overline{b_i^0} \vee \overline{a_i^1}\,a_i^0\,\overline{b_i^1}\,b_i^0 \vee \overline{a_i^1}\,\overline{a_i^0}\,b_i^1\,\overline{b_i^0}, \quad (10)$$

$$c_i^0 = \overline{a_i^1}\,a_i^0\,\overline{b_i^1}\,\overline{b_i^0} \vee \overline{a_i^1}\,a_i^0\,b_i^1\,b_i^0 \vee a_i^1\,\overline{a_i^0}\,b_i^1\,\overline{b_i^0}, \quad (11)$$

– для операции $(a_i \otimes b_i)_{\mathrm{mod}3}$:

$$c_i^1 = a_i^1\overline{a_i^0}\,\overline{b_i^1}\,b_i^0 \vee \overline{a_i^1}\,a_i^0\,b_i^1\,\overline{b_i^0}, \quad (12)$$

$$c_i^0 = \overline{a_i^1}\,a_i^0\,\overline{b_i^1}\,b_i^0 \vee a_i^1\,\overline{a_i^0}\,b_i^1\,\overline{b_i^0}, \quad (13)$$

– для операции $(a_i \times 2)_{\mathrm{mod}3}$:

$$c_i^1 = \overline{a_i^1}\,a_i^0, \quad (14)$$

$$c_i^0 = a_i^1\,\overline{a_i^0}. \quad (15)$$

Очевидно, что функции (10–13) требуется минимизировать, а функции (14, 15) – не нужно. Для синтеза минимальных дизъюнктивных нормальных форм (МДНФ) построим соответствующие карты Карно (таблицы 3, 4). Карты дополнены неопределенными значениями («–») там, где значения СДНФ (см. таблицу 2) не определены. Карта Карно – это альтернативная форма представления таблицы истинности, которая позволяет механизировать способ минимизации логических функций без применения алгебраических средств [22].

Стоит отметить, что существуют и другие методы минимизации логических функций, например, метод непосредственных преобразований, метод неопределенных коэффициентов, различные аналитические методы, диаграммы Вейча и др.

Метод карт Карно основывается на табличном представлении логических функций. Он используется для ручной минимизации функций с числом переменных, не превышающих шести. Данный метод выбран ввиду своей простоты и наглядности.

ТАБЛИЦА 3. Карта Карно для операции сложения в поле *GF*(3^3)

TABLE 3. Karnaugh Map for the Addition Operation in the Field GF(3^3)

$(a_i \oplus b_i)_{\mathrm{mod}3}$												
c_i^1	b_i	**0**	**1**	**3**	**2**		c_i^0	b_i	**0**	**1**	**3**	**2**
a_i	$b_i^1 b_i^0, a_i^1 a_i^0$	00	01	11	10		a_i	$b_i^1 b_i^0, a_i^1 a_i^0$	00	01	11	10
0	00	0	0	–	1		**0**	00	0	1	–	0
1	01	0	1	–	0		**1**	01	1	0	–	0
3	11	–	–	–	–		**3**	11	–	–	–	–
2	10	1	0	–	0		**2**	10	0	0	–	1

ТАБЛИЦА 4. Карта Карно для операции умножения в поле *GF*(3^3)

TABLE 4. Karnaugh Map for the Multiplication Operation in the Field GF(3^3)

$(a_i \otimes b_i)_{\mathrm{mod}3}$												
c_i^1	b_i	**0**	**1**	**3**	**2**		c_i^0	b_i	**0**	**1**	**3**	**2**
a_i	$b_i^1 b_i^0, a_i^1 a_i^0$	00	01	11	10		a_i	$b_i^1 b_i^0, a_i^1 a_i^0$	00	01	11	10
0	00	0	0	–	0		**0**	00	0	0	–	0
1	01	0	0	–	1		**1**	01	0	1	–	0
3	11	–	–	–	–		**3**	11	–	–	–	–
2	10	0	1	–	0		**2**	10	0	0	–	1

Сумма полученных произведений образует МДНФ:

$$c_i^1 = a_i^0 b_i^0 \vee \overline{a_i^1 a_i^0} b_i^1 \vee a_i^1 \overline{b_i^1 b_i^0}, \quad (16)$$

$$c_i^0 = a_i^1 b_i^1 \vee \overline{a_i^1 a_i^0} b_i^0 \vee a_i^0 \overline{b_i^1 b_i^0}, \quad (17)$$

$$c_i^1 = a_i^0 b_i^1 \vee a_i^1 b_i^0, \quad (18)$$

$$c_i^0 = a_i^0 b_i^0 \vee a_i^1 b_i^1. \quad (19)$$

На основе выражений (14–19) получим следующие схемы для булева базиса «И»–«ИЛИ»–«НЕ» (рисунки 5–7).

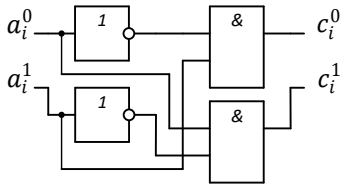

Рис. 5. Функциональная схема умножения произвольных элементов поля *GF*(3³) на 2

Fig. 5. Functional Diagram for Multiplying Arbitrary Elements of the Field GF(3³) *by 2*

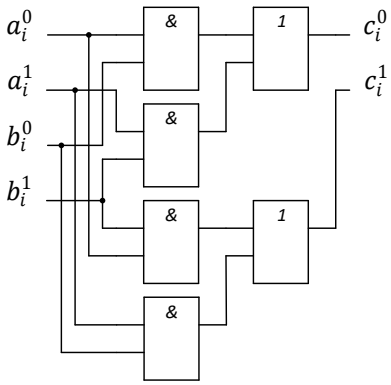

Рис. 6. Структурная схема умножения произвольных элементов поля *GF*(3³)

Fig. 6. Block Diagram of Multiplication of Arbitrary Elements of the Field GF(3³)

В таблице 5 приведены значения цены по Квайну и ранга схем, показанных на рисунках 5–7 [23]. За сложность по Квайну принято суммарное количество входов логических элементов. Рангом схемы обозначено максимальное количество логических элементов, через которые проходит сигнал от выхода ко входу.

ТАБЛИЦА 5. Значения цены по Квайну и ранга для логических схем на рисунках 5–7

TABLE 5. Quine Price and Rank Values for the Logic Circuits in Figures 5–7

Выражение	Цена по Квайну	Ранг схемы
$(a_i \otimes 2)_{\mathrm{mod}3}$	6	2
$(a_i \otimes b_i)_{\mathrm{mod}3}$	12	2
$(a_i \oplus b_i)_{\mathrm{mod}3}$	26	3

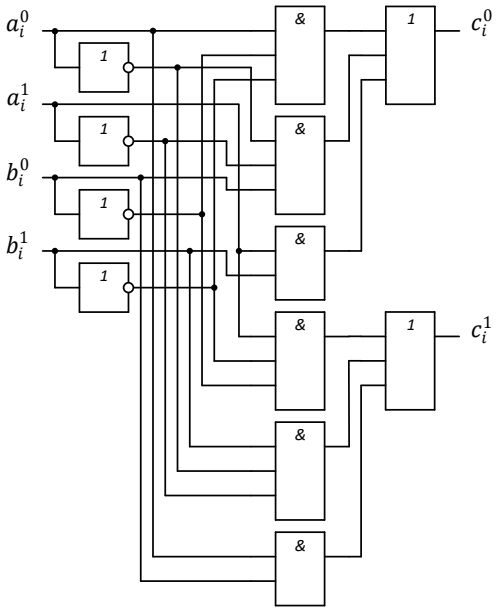

Рис. 7. Структурная схема сложения произвольных элементов поля *GF*(3³)

Fig. 7. Block Diagram of the Addition of Arbitrary Elements of the Field GF(3³)

Метод на основе разложения многозначного расширенного поля Галуа

Анализ схем (рисунки 2 и 4) позволяет сделать вывод, что этап умножения вектора b_i на составную матрицу A может быть оптимизирован. Неприводимые многочлены $P(x)$ в полях $GF(q^k)$ для $q \geq 3$ задают M-последовательности, обладающие свойством деления на подпоследовательности длины $(q^k - 1)/(q - 1)$, связанные друг с другом коэффициентом пропорциональности λ (λ – первообразный элемент поля $GF(q)$) [2, 24].

Например, из таблицы 1 видно, что для $GF(3^3)$ по основанию полинома $P(x) = x^3 + 2x + 1$ элементы $\varepsilon^{13}, \varepsilon^{14}, \ldots, \varepsilon^{25}$ могут быть получены из элементов $\varepsilon^0, \varepsilon^1, \ldots, \varepsilon^{12}$ умножением их на коэффициент $\lambda = 2$.

Таким образом, поле $GF(q^k)$ может рассматриваться как совокупность $(q - 1)$ подмножеств $\{M_0 \quad M_1 \quad \cdots \quad M_{q-2}\}$ размера $(q^k - 1)/(q - 1)$ каждое (20), где: $M_0, M_1, \ldots, M_r, \ldots, M_{q-2}$ – подмножества, состоящие из $(q^k - 1)/(q - 1)$ подряд следующих элементов поля $GF(q^k)$, с номерами $r, r + 1, \ldots, r + (q^k - 1)/(q - 1)$ $(r = 0,1, \ldots, q - 2)$.

Анализ матричного способа умножения произвольных элементов $a = (a_0 \quad \ldots \quad a_{k-1}) = \varepsilon^i$ и $b = (b_0 \quad \ldots \quad b_{k-1}) = \varepsilon^j$ поля $GF(q^k)$ для левого степенного базиса позволяет сформулировать [2] выражение (21), соответствующее функциональной схеме умножения произвольных элементов поля $GF(q^k)$, которая показана в качестве примера на рисунке 4. Из (21) следует выражение (22), где слагаемые c_l^h элементов c_l вектора c являются произведениями скаляров b_h и $a_l \varepsilon^h$ $(l, h = 0, \ldots, k - 1)$ в поле $GF(q)$.

$$GF(q^k) = \begin{Bmatrix} \varepsilon^0 \\ \varepsilon^1 \\ \vdots \\ \varepsilon^{q^k-2} \end{Bmatrix} = \left\{ \begin{Bmatrix} \lambda^0\varepsilon^0 \\ \lambda^0\varepsilon^1 \\ \vdots \\ \lambda^0\varepsilon^{\frac{q^k-1}{q-1}-1} \end{Bmatrix} \begin{Bmatrix} \lambda^1\varepsilon^0 \\ \lambda^1\varepsilon^1 \\ \vdots \\ \lambda^1\varepsilon^{\frac{q^k-1}{q-1}-1} \end{Bmatrix} \cdots \begin{Bmatrix} \lambda^{q-2}\varepsilon^0 \\ \lambda^{q-2}\varepsilon^1 \\ \vdots \\ \lambda^{q-2}\varepsilon^{\frac{q^k-1}{q-1}-1} \end{Bmatrix} \right\} = \begin{Bmatrix} M_0 \\ M_1 \\ \cdots \\ M_{q-2} \end{Bmatrix} = \begin{Bmatrix} \lambda^0 M_0 \\ \lambda^1 M_0 \\ \cdots \\ \lambda^{q-2} M_0 \end{Bmatrix}. \quad (20)$$

$$\begin{gathered} c = (c_0 \quad \dots \quad c_{k-1}) = \varepsilon^{i+j} = \{ab\}_{GF(q^k)} = \{aF^j\}_{GF(q)} = \\ = \{(a_0 \quad \dots \quad a_{k-1})(b_0F^0 + b_1F^1 + \cdots + b_{k-1}F^{k-1})\}_{GF(q)} = \\ = \{b_0(a_0\varepsilon^0 \quad \dots \quad a_{k-1}\varepsilon^{k-1}) + \cdots + b_{k-1}(a_0\varepsilon^{k-1} \quad \dots \quad a_{k-1}\varepsilon^{2k-2})\}_{GF(q)}. \end{gathered} \quad (21)$$

$$\begin{gathered} c_0 = \{b_0a_0\varepsilon^0 + b_1a_0\varepsilon^1 + \cdots + b_{k-1}a_0\varepsilon^{k-1}\}_{GF(q)}, \quad c_1 = \{b_0a_1\varepsilon^1 + b_1a_1\varepsilon^2 + \cdots + b_{k-1}a_1\varepsilon^k\}_{GF(q)}, \dots \\ \dots, c_{k-1} = \{b_0a_{k-1}\varepsilon^{k-1} + b_1a_{k-1}\varepsilon^k + \cdots + b_{k-1}a_{k-1}\varepsilon^{2k-2}\}_{GF(q)}. \end{gathered} \quad (22)$$

Учитывая свойство (20), слагаемое c_l^h может быть представлено следующим образом:

$$\begin{gathered} c_l^h = i(b_h, \lambda) \begin{pmatrix} a_l\{\varepsilon^{h_0}\}_{M_0} \\ a_l\{\varepsilon^{h_1}\}_{M_1} \\ \cdots \\ a_l\{\varepsilon^{h_{q-2}}\}_{M_{q-2}} \end{pmatrix} = \\ = i(b_h, \lambda) \begin{pmatrix} \lambda^0 a_l\{\varepsilon^h\}_{M_0} \\ \lambda^1 a_l\{\varepsilon^h\}_{M_0} \\ \cdots \\ \lambda^{q-2} a_l\{\varepsilon^h\}_{M_0} \end{pmatrix}. \end{gathered} \quad (23)$$

где $l, h, h_0, h_1, \dots h_{q-2} = 0, \dots, k-1$, $i(b_h, \lambda) = (i_0, i_1, \dots, i_{q-2})$ – строка-индикатор, элементы которой определяются в соответствии со следующим индикаторным преобразованием:

$$\begin{gathered} (i_0, \dots, i_j, \dots, i_{q-2}) = \\ = \begin{cases} i_0 = i_1 = \cdots = i_{q-2} = 0, \text{если } b_h = 0, \\ i_j = \begin{cases} 1, \text{если } b_h = \lambda^j \\ 0, \text{если } b_h \neq \lambda^j \end{cases}, \text{если } b_h \neq 0. \end{cases} \end{gathered}$$

Пример 5. Индикаторное преобразование вектора в поле GF(5³)

Поле $GF(5^3)$ по основанию любого примитивного полинома в соответствии со свойством (20) может рассматриваться как совокупность четырех подмножеств. Одним из примитивных элементов поля $GF(5)$ является число $\lambda = 3$ ($\lambda^0 = 1$, $\lambda^1 = 3$, $\lambda^2 = 4$, $\lambda^3 = 2$).

Следовательно, если поле $GF(5^3)$ по основанию соответствующего полинома делится на подмножества по элементу $\lambda = 3$, то значениям элементов b_h вектора b будут соответствовать следующие строки-индикаторы $i(b_h, \lambda)$:

$$i(0,5) = (0,0,0,0), i(1,5) = (1,0,0,0),$$
$$i(2,5) = (0,0,0,1), i(3,5) = (0,1,0,0), i(4,5) = (0,0,1,0).$$

Другим примитивным элементом поля $GF(5)$ является число $\lambda = 2$. Для случая разбиения поля $GF(5^3)$ на подмножества по элементу $\lambda = 2$ индикаторные вектора вычисляются соответствующим образом.

Таким образом, предлагаемый метод (20, 23) позволяет при вычислении полевой свертки перейти от операции умножения произвольных элементов по модулю q к операции индикаторного выбора, которая при синтезе логической схемы может быть реализована с помощью элемента «И». Такой переход требует реализации в синтезируемой схеме, кроме матриц $E = F^0 = \lambda^0F^0$, $F^1 = \lambda^0F^1$, $F^2 = \lambda^0F^2$, …, $F^{k-1} = \lambda^0F^{k-1}$, еще и матриц λ^1F^0, λ^1F^1, …, λ^1F^{k-1}, λ^2F^0, λ^2F^1, …, λ^2F^{k-1}, …, $\lambda^{q-2}F^0$, $\lambda^{q-2}F^1$, …, $\lambda^{q-2}F^{k-1}$. Стоит отметить, что такой подход при должной оптимизации приводит лишь к $q-1$ кратному росту числа умножителей элементов поля $GF(q)$ на константу. Пример функциональной схемы умножения произвольных элементов поля $GF(3^3)$ показан на рисунке 8. Исходные данные взяты из примеров 1 и 2. Примитивный элемент поля $GF(3)$ – $\lambda = 2$. Строки-индикаторы $i(0,2) = (0,0)$, $i(1,2) = (1,0)$, $i(2,2) = (0,1)$ получены в соответствии с выражением (23).

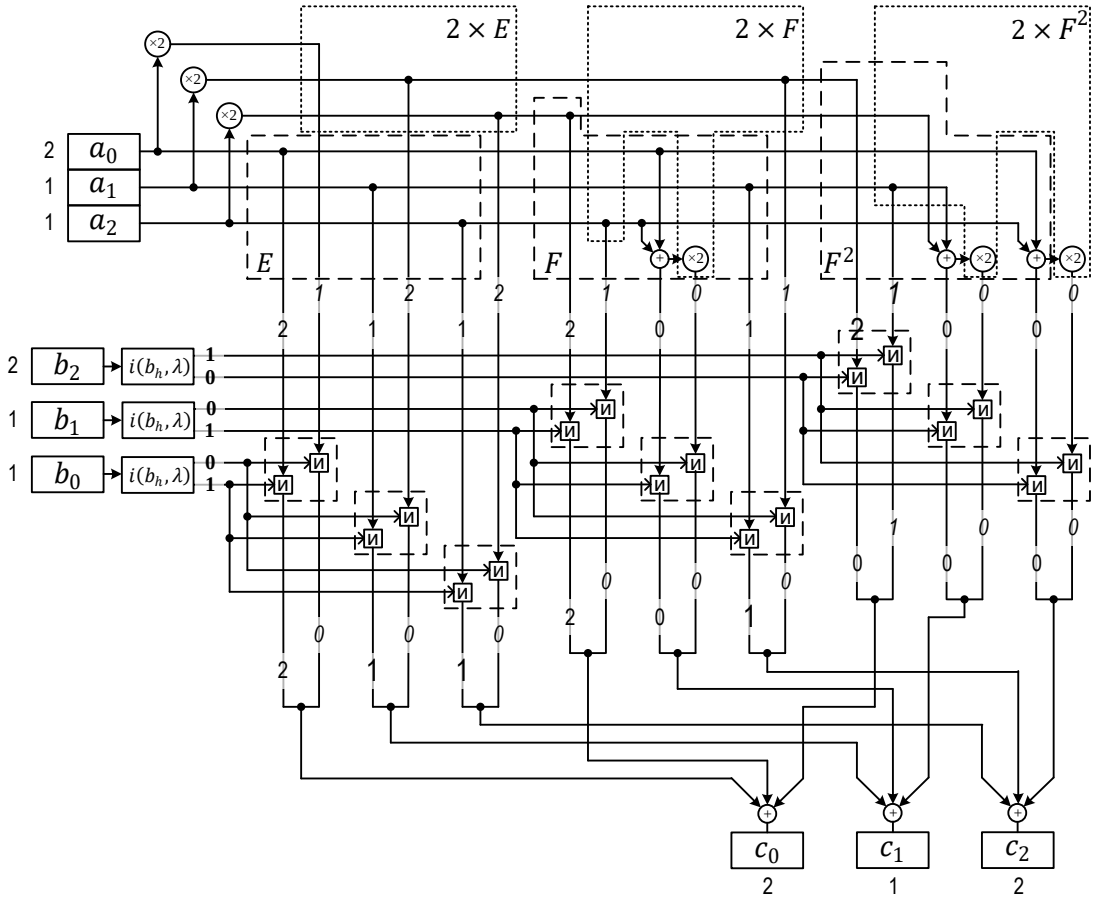

Рис. 8. Функциональная схема предлагаемого метода умножения произвольных элементов поля *GF*(3³)

Fig. 8. Functional Diagram for the Proposed Method of Multiplying Arbitrary Elements from the Field GF(3³)

Предлагаемая схема полевой свертки (см. рисунок 8) построена по аналогии со схемой, реализующей матричный метод (см. рисунок 4), и работает следующим образом. Элемент поля $GF(3^3)$ $a = (a_0 \quad a_1 \quad a_2) = \varepsilon^{11} = (2 \quad 1 \quad 1)$ умножается на матрицы $E, F, F^2, 2 \times E, 2 \times F, 2 \times F^2$.

Из векторов:

$$a \times E = (2 \quad 1 \quad 1),\ a \times 2 \times E = (1 \quad 2 \quad 2),$$
$$a \times F = (2 \quad 0 \quad 1),\ a \times 2 \times F = (1 \quad 0 \quad 1),$$
$$a \times F^2 = (2 \quad 0 \quad 0),\ a \times 2 \times F^2 = (1 \quad 0 \quad 0)$$

с помощью строк-индикаторов:

$$i(b_0 = 1,2) = (1,0), i(b_0 = 1,2) = (1,0),$$
$$i(b_0 = 2,2) = (0,1),$$

полученных из вектора $b = \varepsilon^{20} = (1 \quad 1 \quad 2)$, выбираются вектора:

$$(2 \quad 1 \quad 1), (2 \quad 0 \quad 1) \text{ и } (1 \quad 0 \quad 0),$$

которые затем поступают на соответствующие поразрядные сумматоры. На выходе сумматоров формируется искомый вектор $c = \varepsilon^5 = (2 \quad 1 \quad 2)$.

Представленные на рисунках 4 и 8 схемы являются функциональными. При построении на их основе логических схем (БИС или ПЛИС) предполагается, что элементы $\{a_l\}_{GF(q)}$ вектора a будут заменяться двоичными векторами ($GF(2)$) в соответствии с таблицами истинности, построенными по аналогии с таблицей 2. Операции в поле $GF(q)$ (умножение произвольных элементов, умножение произвольного элемента на константу и сложение произвольных элементов) реализуются в поле $GF(2)$ (в логическом базисе «И»–«ИЛИ»–«НЕ») на основе выражений, которые могут быть получены способом, аналогичным способу вывода выражений (10–15), и минимизированы соответствующим образом. В результате могут быть синтезированы соответствующие логические схемы по аналогии с рисунками 5–7.

Стоит отметить, что выражения, аналогичные выражениям (20–23), могут быть выведены на основании ГТ-преобразования. Блок-схема предлагаемого метода с их использованием показана на рисунке 9.

Ввиду аналогичности матричного метода и метода на основе ГТ-преобразования, разработанный метод целесообразно сравнивать с одним из них, например, с матричным методом. Для этого вна-

чале оценим требуемое количество операций в поле $GF(q)$ для элементов векторов в поле $GF(q^k)$. На основе обобщения функциональной схемы (см. рисунок 8) для любых q и k получена следующая сравнительная оценка количества логических операций в поле $GF(q)$ (таблица 6).

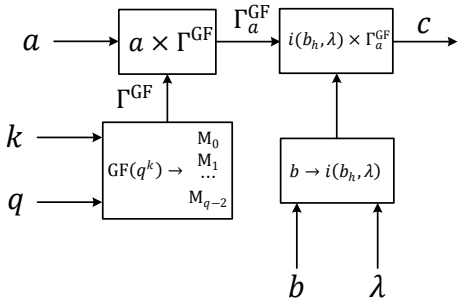

Рис. 9. Блок-схема предлагаемого метода умножения элементов поля $GF(q^k)$ на основе преобразования Ганкеля – Теплица

Fig. 9. Block Diagram for the Proposed Method of Multiplying Elements from the Field GF(qk) Based on the Hankel – Toeplitz Transform

ТАБЛИЦА 6. Сравнительная оценка количества логических операций в поле $GF(q^k)$ для матричного и предложенного методов

TABLE 6. Comparative Estimation for the Number of Logical Operations in the $GF(q^k)$ Field for the Matrix and Proposed Methods

Метод	$GF(q)$			$GF(2)$
	$a_i \otimes \text{const}$	$a_i \otimes b_i$	$a_i \oplus b_i$	«И» (двухвходовый)
Матричный [2]	$\leq (q-2)k$	k^2	$\geq k^2$	–
Разработанный	$\leq 2(q-2)k$	–	$\geq k^2$	$(q-1)k^2$

Из таблицы 6 видно, что разработанный метод рационально использовать для полевой свертки векторов поля $GF(q^k)$ в случаях, когда $k \gg q$. Тогда выигрыш от сокращения числа умножителей превзойдет проигрыш от роста количества сумматоров по модулю q.

В таблице 7 представлены сравнительные оценки стоимости по Квайну и ранга функциональных схем для различных значений k при $q = 3$, полученных на основании следующих выражений:

– матричный метод:

$$6(q-2)k + 12k^2 + 26k^2;$$

– разработанный метод:

$$12(q-2)k + 26k^2 + 2(q-1)k^2.$$

Ранг схемы: матричный метод – $2 + 3 + 2 + 3$; разработанный метод – $2 + 3 + 1 + 3$. Для построения выражений оценки стоимости по Квайну количественные значения стоимости из таблицы 5 для соответствующей операции в поле $GF(q)$ умножены на количество таких операций из таблицы 6. Затем выполнена сумма по всем указанным операциям для соответствующего метода полевой свертки. Ранг схемы вычислен по максимальному пути: вначале определена цепочка логических элементов, затем для каждого элемента подставлено значения ранга из таблицы 5, в конце – выполнена сумма.

ТАБЛИЦА 7. Сравнительная оценка цены по Квайну и ранга схемы операции полевой свертки в поле $GF(3^k)$ для матричного и предложенного методов

TABLE 7. Comparative Evaluation of the Quine Price and Rank of the Field Convolution Operation Scheme in the $GF(3^k)$ Field for the Matrix and Proposed Methods

k	Цена по Квайну			Ранг схемы		
	M_m	M_{GF}	W, %	M_m	M_{GF}	W, %
3	360	306	15,00	33	32	3,03
30	34 380	27 360	20,42			
300	3 421 800	2 703 600	20,99			
3 000	342 018 000	270 036 000	21,05			
30 000	34 200 180 000	27 000 360 000	21,05			
300 000	3 420 001 800 000	2 700 003 600 000	21,05			

Усл. обозначения: M_m – матричный метод; M_{GF} – разработанный метод; W – выигрыш

Из таблицы 7 видно, что для больших k формируется устоявшееся соотношение цены по Квайну, разработанное к матричному методу, равное 0,7895. Таким образом для $q = 3$ и $k \gg q$ выигрыш составляет ≈ 21,05 %. Из анализа таблицы 6 можно предположить, что с ростом q выигрыш снизится. Анализ зависимости выигрыша от q требует дополнительного изучения, а именно синтеза выражений и функциональных схем для операций в соответствующих полях Галуа.

Заключение

Проведенный анализ однотактных методов умножения элементов расширенного поля Галуа позволил выявить целесообразность поиска оптимальных методов вычисления полевых сверток. Логарифмирование–антилогарифмирование имеет ограниченную область применения для больших значений q и k.

Разработан метод вычисления полевой свертки на основе использования свойства разложения многозначного расширенного поля Галуа на подмножества по степени примитивного элемента соответствующего простого поля с использованием индикаторного представления одного из множителей. Доказано сокращение общего числа логических элементов и как следствие цены по Квайну и ранга схемы. Приведены примеры вычисления полевых сверток в многозначных полях Галуа различными методами. Дана сравнительная оценка разработанного метода.

Список источников

1. Рахман П.А. Кодирование информации с применением кодов Рида-Соломона. Уфа: УГНТУ, 2012. 167 с.
2. Когновицкий О.С. Теория, методы и алгоритмы решения задач в телекоммуникациях на основе двойственного базиса и рекуррентных последовательностей. Дис. … докт. техн. наук. СПб.: СПбГУТ, 2011. 427 с. EDN:QFKYXL
3. Муттер В.М. Основы помехоустойчивой телепередачи информации. Л.: Энергоатомиздат. Ленингр. отд-ние, 1990. 288 с.
4. Когновицкий О.С., Сюрин В.Н., Ассанович Б.А. Метод вычисления логарифма в конечном поле GF(2^m) // Девятая всесоюзная конференция по теории кодирования и передачи информации. Часть 1. Тезисы докладов. Одесса, 1988. С. 100–102.
5. Рахман П.А. Рекуррентный алгоритм вычисления усеченной свертки полиномов над полем Галуа и его аппаратная реализация // Международный журнал прикладных и фундаментальных исследований. 2015. № 12-2. С. 231–235. EDN:SZAEYV
6. Иванова И.В. Научные основы создания автоматизированных систем кодирования данных в конечных полях Галуа методами дискретной алгебры Клини. Дис. … докт. техн. наук. СПб.: СЗТУ, 2005. 276 с. EDN:NNZFPD
7. Берлекэмп Э.Р. Алгебраическая теория кодирования. Пер. с англ. М.: Мир, 1971. 478 с.
8. Мак-Вильяме Ф.Дж., Споэн Н.Дж.А. Теория кодов, исправляющих ошибки. Пер. с англ. М: Связь, 1979. 744 с.
9. Касами Т., Токура Н., Ивадари Е., Ирагаки Я. Теория кодирования. М.: Мир, 1978. 576 с.
10. Рахман П.А. Арифметика двоичного поля Галуа на базе быстрого умножения и инвертирования элементов поля и ее аппаратная реализация // Международный журнал прикладных и фундаментальных исследований. 2015. № 12-3. С. 403–408. EDN:VBUMBJ
11. Листопад Е.В., Петровский А.А. Особенности реализации операций умножения элементов поля Галуа на FPGA // 53-я научная конференция аспирантов, магистров и студентов БГУИР (Минск, Республика Беларусь, 2–6 мая 2017 г.). Минск: Белорусский государственный университет информатики и радиоэлектроники, 2017. С. 234–235.
12. Касперски К. Полиномиальная арифметика и поля Галуа**,** или информация, воскресшая из пепла II // Системный администратор. 2003;10(11):84–90. EDN:RDELQN
13. Салимов Г.Ю. Предложения по реализации умножения в поле Галуа над неприводимым многочленом на примере преобразования *L* в алгоритме ГОСТ Р 34.1 2 2015 // XXII научно-практическая конференция «РусКрипто 2020» (17–29 марта 2020 г.) 2020. URL: https://ruscrypto.ru/resource/archive/rc2020/files/14_salimov.pdf (дата обращения 17.04.2025)
14. Лидл Р., Нидеррайтер Г. Конечные поля. Пер. с англ. М.: Мир, 1988. 818 с.
15. Габидулин Э.М., Афанасьев В.Б. Кодирование в радиоэлектронике. М.: Радио и связь, 1986. 176 с.
16. Питерсон Ч., Уэлдон Э. Коды, исправляющие ошибки. Пер. с англ. М.: Мир, 1976. 594 с.
17. Кларк Дж., мл., Кейн Дж. Кодирование с исправлением ошибок в системах цифровой связи. Пер. с англ. М.: Радио и связь, 1987. 391 с.
18. Золотарев В.В. Теория и алгоритмы многопорогового декодирования. М.: Радио и связь, 2006. 266 с.
19. Трифонов П.В. Основы помехоустойчивого кодирования. СПб.: Университет ИТМО, 2022. 231 с.
20. Ишмухаметов Ш.Т., Латыпов Р.Х., Столов Е.Л., Рубцова Р.Г. Введение в теорию чисел и теорию кодирования: учебное пособие. Казань: Казанский университет, 2014. 65 с.
21. Ишмухаметов Ш.Т., Рубцова Р.Г. Вычисления в конечных полях: учебно-методическое пособие. Казань: Казанский университет, 2019. 23 с.
22. Назарова А.К., Локтионова О.Е., Спесивцев Г.А. Карты Карно // Международная научно-практическая конференция «Моделирование информационных систем и технологий» (Воронеж, Российская Федерация, 02 апреля 2024 г.). Воронеж: Воронежский государственный лесотехнический университет имени Г.Ф. Морозова, 2024. С. 116–121. EDN:JPIWIW
23. Исмагилова Е.И. Булевы функции и построение логических схем. М.: МИРЭА, 2015. 160 с.
24. Цирлер Н. Линейные возвратные последовательности // Кибернетический сборник. 1963. № 6. С. 31–48.

References

1. Rakhman P. *Information Coding Using Reed-Solomon Codes*. Ufa: Ufa State Petroleum Technical University Publ.; 2012, 167 p. (in Russ.)
2. Kognovitsky O. *Theory, Methods and Algorithms for Solving Problems in Telecommunications Based on a Dual Basis and Recurrent Sequences*. D.Sc Thesis. St. Petersburg: The Bonch-Bruevich Saint-Petersburg State University of Telecommunications Publ.; 2011. 427 p. (in Russ.) EDN:QFKYXL
3. Mutter V. *Fundamentals of Noise-Immune Television Transmission of Information*. Leningrad: Energoatomizdat; 1990. 288 p. (in Russ.)
4. Kognovitsky O.S., Syurin V.N., Assanovich B.A. Method for calculating logarithms in the traditional field GF(2m). *Proceedings of the 9th All-Union Conference on Theories of Coding and Transmission of Information. Part 1*. Odessa; 1988. p.100–102. (in Russ.)
5. Rakhman P.A. Algorithm for Computing the Truncated Convolution of the Polynomials Over Galois Field and Its Hardware Implementation. *Mezhdunarodnyi zhurnal prikladnykh i fundamentalnykh issledovanii*. 2015;12-2:231–235. (in Russ.) EDN:SZAEYV
6. Ivanova I.V. *Scientific Foundations for Creating Automated Data Coding Systems in Finite Galois Fields Using Discrete Kleene Algebra Methods*. D.Sc Thesis. St. Petersburg: North-West Technical University Publ; 2005. 276 p. (in Russ.) EDN:NNZFPD

7. Berlekamp E. R. *Algebraic Coding Theory*. 1968.
8. McWilliam F.J., Sloane N.J.A. *The theory of error-correcting codes.* North-Holland Publishing; 1977.
9. Kasami T., Tokura N., Iwadari E., Iragaki Y. *Coding Theory*. Moscow: Mir Publ.; 1978. 576 p. (in Russ.)
10. Rahman P.A. Arithmetic of a binary Galois field based on fast multiplication and inversion of field elements and its hardware implementation. *Mezhdunarodnyi zhurnal prikladnykh i fundamentalnykh issledovanii*. 2015;12-3:403–408. EDN:VBUMBJ
11. Listopad E.V., Petrovsky A.A. Features of the implementation of operations of multiplication of Galois field elements on FPGA. *Proceedings of the 53rd Scientific Conference of Graduate Students, Masters and Students of BSUIR, 2–6 May 2017, Minsk, Republic of Belarus*. Minsk: Belarusian State University of Informatics and Radioelectronics Publ.; 2017. p.234–235. (in Russ.)
12. Kasperski K. Polynomial arithmetic and Galois fields, or information resurrected from the ashes II. *Sistemnyi administrator*. 2003;10(11):84–90. (in Russ.) EDN:RDELQN
13. Salimov G.Yu. Proposals for the implementation of multiplication in a Galois field over an irreducible polynomial using the example of transformation L in the GOST R 34.1 2 2015 algorithm. *Proceedings of the XXIInd Scientific and Practical Conference "RusCrypto 2020", 17–29 March 2020*. 2020. (in Russ.) URL: https://ruscrypto.ru/resource/archive/rc2020/files/14_salimov.pdf [Accessed 17.04.2025]
14. Lidl R., Niederreiter G. *Finite fields*. London: Addison-Wesley Publishing; 1983.
15. Gabidullin E.M., Afanasyev V.B. *Coding in Radio Electronics.* Moscow: Radio i sviaz Publ.; 1986. 176 p. (in Russ.)
16. Peterson W.W., Weldon-jr E.J. *Error-Correcting Codes*. MIT Press; 1972. 575 p.
17. Clark J., Jr., Kane J. *Error-Correcting Coding for Digital Communication*. New York: Plenum Press; 1981.
18. Zolotarev V.V. *Theory and Algorithms of Multithreshold Decoding*. Moscow: Radio i sviaz Publ.; 2006. 266 p. (in Russ.)
19. Trifonov P.V. *Fundamentals of Noise-Corrective Coding*. St. Petersburg: ITMO University Publ.; 2022. 231 p. (in Russ.)
20. Ishmukhametov Sh.T., Latypov R.Kh., Stolov E.L., Rubtsova R.G. *Introduction to Number Theory and Coding Theory*. Kazan: Kazan University Publ.; 2014. 65 p. (in Russ.)
21. Ishmukhametov Sh.T., Rubtsova R.G. *Calculations in Finite Fields*. Kazan: Kazan University Publ.; 2019. 23 p. (in Russ.)
22. Nazarova A., Loktionova O., Spesivtsev G. Carnot Cards. *Proceedings of the* International Scientific and Practical Conference on Modeling of Information Systems and Technologies, 02 April 2024, Voronezh, Russian Federation. Voronezh: Voronezh State Forestry University Publ.; 2024. p.116–121. (in Russ.) EDN:JPIWIW
23. Ismagilova E.I. *Boolean Functions and Construction of Logical Circuits*. Moscow: MIREA – Russian Technological University Publ.; 2015. 160 p.
24. Zierler N. Linear vecurring sequences. *Journal of the Society for Industrial and Applied Mathematics.* 1959;7(1):31–48. (in Russ.)

Статья поступила в редакцию 09.12.2024; одобрена после рецензирования 19.03.2025; принята к публикации 20.03.2025.
The article was submitted 09.12.2024; approved after reviewing 19.03.2025; accepted for publication 20.03.2025.

Информация об авторе:

УЛЬЯНОВ Илья Владимирович
кандидат технических наук, сотрудник Академии Федеральной службы охраны Российской Федерации
https://orcid.org/0009-0009-7607-7868

Автор сообщает об отсутствии конфликтов интересов.
The author declares no conflicts of interests.