НАО «Алматинский университет энергетики и связи имени Гумарбека Даукеева»

|  |
| --- |
| УДК 621.396.6:004.056.5+623.36 |

**ВИТУЛЁВА ЕЛИЗАВЕТА СЕРГЕЕВНА**

**Постиндустриальная парадигма развития инфокоммуникационного сегмента оборонно-промышленного комплекса РК**

6D071900 – Радиотехника, электроника и телекоммуникации

Диссертация в форме серии статей

на соискание степени доктора философии (PhD)

Научный консультант:

кандидат технических наук,

профессор Коньшин С.В.

Зарубежный научный консультант:

доктор философских наук,

профессор Габриелян О.А.

Республика Казахстан

Алматы, 2024

**СОДЕРЖАНИЕ**

|  |  |
| --- | --- |
| ВВЕДЕНИЕ…………………………………………………………………….. | 4 |
| ТЕКСТЫ ПУБЛИКАЦИЙ, ПРЕДСТАВЛЕННЫХ НА СОИСКАНИЕ СТЕПЕНИ ДОКТОРА ФИЛОСОФИИ (PHD)………………………………... | 6 |
| 1 Повышение эффективности использования инструменты многозначной логики…………………………………………………………………………... | 6 |
| 1.1 Аннотация первой статьи………………………………………………… | 6 |
| 1.2 Введение первой статьи………………………………………………….. | 6 |
| 1.3 Визуализация переменных многозначной логики………………………. | 8 |
| 1.4 Сведение операций многозначной логики к алгебраическим…………. | 11 |
| 1.5 Вычислительная реализация семизначных логических операций…….. | 16 |
| 1.6 Заключение первой статьи……………………………………………….. | 21 |
| 1.7 Список использованных источников первой статьи……………………. | 22 |
| 2 Особенности цифровой обработки сигналов алгоритмы, использующие поля Галуа GF (2n+1)…………………………………………………………… | 25 |
| 2.1 Аннотация второй статьи………………………………………………… | 25 |
| 2.2 Введение второй статьи………………………………………………….. | 25 |
| 2.3 Преимущества чисел Мерсенна для вычисления систем по модулю целого числа……………………………………………………………………. | 26 |
| 2.4 Алгоритмы вычислений по модулю в частном случае поля GF(17)…… | 28 |
| 2.5 Попеременное двоичное кодирование…………………………………... | 33 |
| 2.6 Алгоритмические основы цифрового логарифма в GF(2n+1) поле……. | 36 |
| 2.7 Выводы второй статьи……………………………………………………. | 38 |
| 2.8 Список использованной литературы второй статьи…………………….. | 39 |
| 3 Особенности поля Галуа GF(257) и его использование для цифровой обработки сигналов……………………………………………………………. | 42 |
| 3.1 Аннотация третьей статьи………………………………………………... | 42 |
| 3.2 Введение третьей статьи…………………………………………………. | 42 |
| 3.3 Методы: сравнение чисел Мерсена и квази-Мерсена, используемых для цифровой обработки сигналов……………………………………………. | 44 |
| 3.4 Результаты и обсуждение третьей статьи……………………………….. | 48 |
| 3.5 Заключение третьей статьи………………………………………………. | 65 |
| 3.6 Список использованной литературы третьей статьи…………………… | 66 |
| 4 Особенности применения частичных преобразований для вычисления редуцированных числовых преобразований…………………………………. | 72 |
| 4.1 Аннотация четвертой статьи……………………………………………... | 72 |
| 4.2 Введение четвертой статьи………………………………………………. | 72 |
| 4.3 Литературный обзор четвертой статьи………………………………….. | 74 |
| 4.4 Постановка проблемы четвертой статьи………………………………… | 76 |
| 4.5 Методы четвертой статьи………………………………………………… | 77 |
| 4.6 Результаты четвертой статьи……………………………………………... | 81 |
| 4.7 Обсуждение четвертой статьи…………………………………………… | 89 |
| 4.8 Выводы четвертой статьи………………………………………………… | 95 |
| 4.9 Список литературы четвертой статьи……………………………………. | 96 |
| ЗАКЛЮЧЕНИЕ………………………………………………………………… | 102 |

**ВВЕДЕНИЕ**

Диссертационная работа на тему «Постиндустриальная парадигма развития инфокоммуникационного сегмента оборонно-промышленного комплекса РК» в форме серии статей представлена на соискание степени доктора философии (PhD) по специальности «6D071900 – Радиотехника, электроника и телекоммуникации».

В работе отражены переводы четырех оригинальных статей, которые освещают основные результаты и выводы проведенных исследований и соответствуют п. 5-1 Правил присуждения степеней, а именно «2) серии из не менее, чем двух статей и одного обзора или трех статей, опубликованных в изданиях, входящих в первый и (или) второй квартиль по импакт-фактору по данным Journal Citation Reports (Жорнал Цитэйшэн Репортс) компании Clarivate Analytics (Кларивэйт Аналитикс) по одной из научных областей, соответствующих содержанию диссертации. В одной из статей докторант является первым автором или автором для корреспонденции».

В данной серии представлены четыре работы, в двух из которых докторант является автором для корреспонденции, что также соответствует п. 5-1 Правил присуждения степеней.

Перечень статей:

1. Suleimenov I.E., Vitulyova Y.S., Kabdushev S.B., Bakirov A.S. Improving the efficiency of using multivalued logic tools. Scientific Reports. – 2023. – № 13(1). – P. 1108.

2. Suleimenov I.E., Vitulyova Y.S., Matrassulova D.K. Features of digital signal processing algorithms using Galois fields GF (2n+1). Plos one. – 2023. – 18(10). – P. e0293294.

3. Bakirov A., Matrassulova D., Vitulyova\* Ye., Shaltykova D., Suleimenov I. The specifics of the Galois field GF(257) and its use for digital signal processing. Scientific Reports. – 2024. – Vol. 14. – P. 15376. (Докторант является автором для корреспонденции).

4. Suleimenov I., Kadyrzhan A., Matrassulova D., Vitulyova\* Y. Peculiarities of Applying Partial Convolutions to the Computation of Reduced Numerical Convolutions. Applied Sciences (Switzerland). – 2024. – № 14(14). – P. 2076-3417. (Докторант является автором для корреспонденции).

Исследования в том числе выполнялись в соответствии с утвержденным планом в рамках научных проектов с непосредственным участием соискателя, что также было отражено в заключительных отчетах проектов:

- AP14870281 «Разработка новых подходов к цифровой обработке изображений с использованием сверточных нейронных сетей» (2022-2024);

- AP14870416 «Разработка новых подходов к решению философских проблем многозначной логики как средства установления закономерностей мышления» (2022-2024);

- «Жас ғалым» AP15473224 «Разработка новых подходов к построению теории научных революций» (2022-2024).

**ТЕКСТЫ ПУБЛИКАЦИЙ, ПРЕДСТАВЛЕННЫХ НА СОИСКАНИЕ СТЕПЕНИ ДОКТОРА ФИЛОСОФИИ (PHD)**

**1 Повышение эффективности использования инструменты многозначной логики**

**1.1 Аннотация первой статьи**

Многозначные логики становятся одним из важнейших инструментов информационных технологий. Они востребованы для создания систем искусственного интеллекта, приближенных к человеческому, поскольку функционирование последнего не может быть сведено к операциям двоичной логики. В то же время остро стоит проблема повышения эффективности использования результатов исследований в области многозначных логик, а также проблема интерпретации переменных многозначной логики. Эти проблемы создают определенные междисциплинарные барьеры и затрудняют внедрение результатов исследований в области многозначных логик в другие области знания. Показано, что проблема интерпретации переменных многозначной логики может быть снята путем установления соответствия с переменными нечеткой логики. Повышение эффективности использования операций многозначных логик и их переменных может быть обеспечено за счет использования их тесной связи с полями Галуа. Эта связь, в частности, позволяет свести любые операции многозначных логик, число переменных в которых равно простому числу, к алгебраическим функциям, аргументы которых принимают значения в полях Галуа. Это позволяет, в частности, устранить весьма громоздкие конструкции, используемые в работах по многозначной логике, и сделать ее аппарат удобным для использования в смежных научных дисциплинах в области информационных технологий. Непосредственная проверка адекватности алгоритмов, основанных на использовании полей Галуа, может быть осуществлена с помощью радиоэлектронных схем, примеры которых представлены в настоящей работе.

**1.2 Введение первой статьи**

Появление неаристотелевских логик (в частности, логики Лукасевича [1], "воображаемой логики" Н. Васильева [2]) в начале XX века, очевидно, было связано с трансформацией общей ситуации в философии математики и дискуссиями по проблемам обоснования математики и логики как таковых [3]. Как отмечалось [2], Н. Васильев предложил проект неаристотелевской логики, построенной без использования закона противоречия, исходя из аналогии с неевклидовой геометрией Н. Лобачевского, исключающей использование пятого постулата Евклида, который также изначально называл свою геометрию "воображаемой". Построение логик, частично или полностью отказывающихся от использования закона исключенного третьего ("каждое утверждение либо истинно, либо ложно", если использовать простейший вариант интерпретации), привело впоследствии к большому разнообразию многозначных логик [4,5], включая параконсистентные логики [6], паракомплексные логики [7] и др.

Де-факто, в настоящее время существует огромное количество разновидностей многозначных логик, но вопрос о том, насколько точно они применимы к описанию законов мышления, остается открытым [8].

В этой связи стоит отметить, что в соответствии с традицией, восходящей к Аристотелю, логика рассматривалась как наука о том, как правильно рассуждать, как наука о законах мышления. Именно так трактовал ее Дж. У. Буль. В известной монографии по истории математики [3] приводится следующий отрывок из одной из главных работ Дж. Буля "Исследование законов мышления", иллюстрирующий именно такой подход, господствовавший тогда в области создания логики:

"В рассматриваемом нами трактате мы намерены исследовать фундаментальные законы тех операций ума, посредством которых осуществляется мышление, чтобы выразить их на символическом языке исчисления и на этой основе построить науку логики и ее метод".

Очевидно, что большинство современных работ по многозначным логикам достаточно далеко отошли от этой традиции, иначе проблема интерпретации многозначных логик и их классификации не стояла бы так остро.

Проблема применимости многозначных логик к отражению законов мышления наиболее тесно связана с проблемой интерпретации переменных многозначной логики, которая остается актуальной и в настоящее время [9,10]. Если в рамках бинарной логики ее переменные могут быть однозначно связаны с понятием истины, то для многозначных логик такая однозначность утрачивается, что определяет актуальность исследований в области философии многозначных логик, которые в настоящее время активно ведутся [11,12].

Однако следует признать, что полноценное возвращение к традиции, рассматривающей логику как отражение законов мышления, очевидно, не может быть осуществлено иначе, чем на междисциплинарной основе. А это, в свою очередь, требует преодоления ярко выраженных междисциплинарных барьеров. Язык, на котором написаны работы по многозначной логике, остается сложным для понимания значительной части специалистов из других областей знания, в частности, из области информационных технологий.

Определенным шагом к преодолению междисциплинарных барьеров является сознательное решение проблемы видимости переменных многозначных логик, затронутой в [13,14].

Для решения этой проблемы целесообразно использовать соответствие между многозначными логиками и алгебраическими структурами, такими как поля Галуа, которые широко используются в современных информационных технологиях, особенно в криптографии [15-17]. Это соответствие проще всего установить, когда число переменных конкретной многозначной логики равно степени простого числа p. В этом случае элемент поля Галуа $GF(p^{n})$ может быть присвоен каждому значению переменной многозначной логики в соответствии один к одному.

Для полей Галуа, в свою очередь, можно предложить следующую иллюстративную интерпретацию. Как подчеркивается [18-19], стандартная модель сигнала — это функция, принимающая значения на множестве вещественных чисел. Однако в случае, когда сигнал сводится к некоторому набору дискретных уровней, укладывающихся в конечный диапазон измерений амплитуды, такой подход не является обязательным. В качестве модели сигнала может быть использована функция, принимающая значения в любой конечной алгебраической структуре, например, в полях Галуа. Простейший вид полей Галуа $GF(p)$ образуется через гомоморфизм кольца целых чисел на кольцо классов вычетов по модулю $p$где $p$ простое число.

В данной работе мы показываем, что проблема интерпретации переменных многозначной логики может быть решена, например, путем установления соответствия между переменными многозначной логики и переменными нечеткой логики. Переменные многозначной логики также могут быть присвоены уровням оцифрованного сигнала в случае, когда моделью сигнала является функция, принимающая значения в полях Галуа. В более широком смысле переменные многозначной логики могут быть интерпретированы через установление связей между понятиями (например, философскими категориями). Во всех этих случаях важно иметь инструмент, позволяющий привести логические отношения к алгебраической форме. Для случая, когда множество переменных многозначной логики может быть отнесено к полю $GF(p^{n})$эта проблема решается с помощью аналога алгебраической нормальной формы, представленного в данной работе.

В разделе 1 показано, что использование многозначных логических переменных может быть сделано явным, в том числе путем отображения на многозначные логические переменные.

В разделе 2 показано, что для случая, когда количество переменных равно простому числу, вместо таблиц истинности, традиционно используемых в работах по многозначной логике, можно также использовать аналог алгебраической нормальной формы (полином Жегалкина).

В разделе 3 приводится конкретный пример, показывающий, что многозначные логические операции могут выполняться с помощью электронных устройств, построенных на типичных компонентах двоичной логики.

**1.3 Визуализация переменных многозначной логики**

Наглядные иллюстрации практического использования переменных многозначных логик проще всего предложить, ориентируясь на подходы, используемые в нечеткой логике. Как известно, нечеткая логика устанавливает определенное соответствие между диапазонами непрерывно изменяющихся параметров и обозначающими их лингвистическими переменными [20]. Упрощенно говоря, аппарат лингвистических переменных позволяет "превратить в слова" значения параметров, которые при определенных условиях могут быть количественно измерены с высокой точностью.

Интересно отметить, что лингвистические переменные были введены в практику задолго до появления нечеткой логики. Например, в морском деле традиционно существует набор команд "полный вперёд, ..., медленный вперёд, ..., медленный вперёд, ..., полный вперёд". Аналогичный вывод справедлив и в отношении компасной розы (рис. 1), которая также традиционно используется в морском деле.



Рисунок 1 **–** Роза компаса на 8 ветров

Рисунок 1 подчеркивает, что 8-элементная роза компаса может быть использована для визуальной интерпретации переменных 9-значной логики.

Переменным такой логики можно поставить в соответствие элементы поля Галуа $GF(3^{2})$которое, в свою очередь, может быть построено как алгебраическое расширение поля $GF(3)$.

Напомним, что метод алгебраических расширений можно рассматривать как обобщение метода, с помощью которого строятся комплексные числа. Продемонстрируем этот факт на простом примере построения поля $GF(3^{2})$.

Поле $GF(3)$ содержит три элемента. Их можно выбрать как $(-1,0,1)$ задав следующие правила сложения.

$1+1=-1; -1-1=1$ (1)

Согласно методу алгебраических расширений, к этому (или любому другому) полю присоединяется дополнительный элемент $θ$который является корнем уравнения, несводимого (не имеющего решений) в данном поле, присоединяется к этому (или любому другому) полю.

$f\left(x\right)=0$ (2)

где $f\left(x\right)$ многочлен степени n, $x$ переменная, принимающая значения в $GF(3)$.

В частном случае, когда $n=2$ таким несводимым уравнением является уравнение, позволяющее построить комплексные числа

$x^{2}+1=0$ (3)

Тогда элемент $θ$ можно рассматривать как логическую мнимую единицу, а элементы поля $GF(3^{2})$ можно представить в виде

$A=a\_{0}+ia\_{1}$, (4)

где переменные $a\_{0},a\_{1}$ относятся к основному полю.

В этом случае мы можем выполнять алгебраические операции с элементами формы (4) по формулам (1) и (3). Например,

$i+i=-i; -i-i=i; -i+i=i-i=0$, (5)

Правила умножения остаются теми же, что и при классическом использовании комплексных чисел, в частности,

$i^{2}=-1; i∙\left(a\_{1}+ia\_{2}\right)=ia\_{1}-a\_{2}$, (6)

Элементы этого поля перечислены в таблице 1.

Таблица 1. Элементы поля Галуа GF(32) в используемом представлении

|  |  |  |  |
| --- | --- | --- | --- |
| $$a$$ | $$a\_{2}=-1$$ | $$a\_{2}=0$$ | $$a\_{2}=1$$ |
| $$a\_{1}=-1$$ | $$-1-i$$ | $$-1$$ | $$-1+i$$ |
| $$a\_{1}=0$$ | $$-i$$ | 0 | $$i$$ |
| $$a\_{1}=1$$ | $$1-i$$ | 1 | $$1+i$$ |

В общем случае любой элемент поля $GF(3^{n})$ может быть представлен в виде линейной комбинации мощностей $θ$.

$A=\sum\_{0}^{n-1}θ^{j}a\_{j}$ (7)

где $θ$ примитивный элемент, $a\_{j}$ коэффициенты из основного поля $GF\left(3\right)$и $n$ степень многочлена $f\left(x\right)$ порождающего элемент $θ$.

Поле $GF(3^{2})$ содержит восемь ненулевых элементов (табл. 1). Используя обозначение (4) в качестве логического представления координат, эти восемь элементов можно отнести к направлениям компасных роз, что показано на рис. 1.

В данном примере существует взаимно-однозначное соответствие между элементами многозначной логики, лингвистическими переменными и элементами поля Галуа. Точнее, элементы розы компаса допускают все вышеперечисленные интерпретации, которые находятся во взаимно однозначном соответствии.

Таким образом, проблема интерпретации многозначных логических переменных может быть снята, если эти переменные интерпретируются через соответствие переменным нечеткой логики. Такой подход, как показано в разделе 3, является обобщающим. А именно, в этой интерпретации вместо переменных нечеткой логики может использоваться достаточно широкий спектр различных терминов (в том числе философских категорий). Очевидно, что смысл словам естественного языка придает не конкретный набор звуков или символов, их обозначающих, а то, что каждое из этих слов встроено в общую структуру языка. Таким образом, значение терминов фактически определяется связями между ними. Оппозиция "истина - ложь", составляющая методологическую основу бинарной логики, является лишь простейшей формой такой связи.

Покажем, что для случая, когда число переменных многозначной логики равно простому числу, любые операции в такой логике могут быть сведены к операциям сложения и умножения в поле Галуа

**1.4 Сведение операций многозначной логики к алгебраическим**

Операции многозначной логики обычно отображаются в виде таблиц истинности. Так, в следующей таблице 2 отражены операции логики парадоксов Г. Приста [21].

Таблица 2. Значения логической функции, соответствующие операциям дизъюнкции и конъюнкции в логике парадоксов Г. Приста

|  |  |  |  |
| --- | --- | --- | --- |
| $$F\left(x,y\right)= ∨$$ | $$y=0$$ | $$y=1$$ | $$y=2$$ |
| $$x=0$$ | 0 | 1 | 2 |
| $$x=1$$ | 1 | 1 | 2 |
| $$x=2$$ | 2 | 2 | 2 |

|  |  |  |  |
| --- | --- | --- | --- |
| $$F\left(x,y\right)= ∧$$ | $$y=0$$ | $$y=1$$ | $$y=2$$ |
| $$x=0$$ | *0* | *0* | *0* |
| $$x=1$$ | *0* | *1* | *1* |
| $$x=2$$ | *0* | *1* | *2* |

В этих таблицах символы "0", "1" и "2" обозначают логические переменные. Интерпретация переменных тернарной логики как "Истина", "Ложь", "Неопределенность" восходит к работам Лукасевича. Интерпретация таких операций (дизъюнкция, конъюнкция, отрицание и т.д.) применительно к троичной логике может быть различной, так же, как и использование конкретных символов в таких таблицах является не более чем вопросом соглашения.

Такое табличное представление не всегда удобно. Операции над логическими переменными, которым приписываются элементы поля Галуа, можно свести к алгебраическим. Для наглядности это можно сделать, например, следующим образом.

Чтобы не загромождать заметки, мы рассмотрим случай произвольной функции $f(x,y)$, принимающей значения в области $GF(p)$где $x,y$ являются элементами одного поля Галуа. Этой функции соответствует таблица истинности, заданная упорядоченным перечислением элементов $f(x\_{i},y\_{j})$, $i,j=0,2…p-1$.

Рассмотрим следующее выражение

$g\_{i}\left(x\right)=1-\left(x-x\_{i}\right)^{p-1}$ (8)

где $x\_{i}$ фиксированный элемент поля $GF(p)$.

Из теории полей Галуа известно, что все ненулевые элементы поля $GF(p)$ являются корнями уравнения

$θ^{p-1}-1=0$ (9)

То есть любой ненулевой элемент поля $GF(p)$возведенный в $p-1$до большой степени, дает единицу.

Следовательно, функции $g\_{i}\left(x\right)$ обладают следующим свойством

$g\_{i}\left(x\right)=\left\{\begin{matrix}1, x=x\_{i}\\0, x\ne x\_{i}\end{matrix}\right.$ (10)

Это позволяет рассматривать их как логический аналогδ -функции.

Образуем следующий многочлен

$F\left(x,y\right)=\sum\_{i,j=0}^{i,j=p-1}f\left(x\_{i},y\_{j}\right)g\_{i}\left(x\right)g\_{j}\left(y\right)$ (11)

где значения $f(x\_{i},y\_{j})$ образуют таблицу истинности, как в таблице 2.

Когда в выражение (11) подставляется конкретная пара $x\_{i\_{0}},y\_{j\_{0}}$ значений логических переменных (точнее, соответствующих им элементов поля Галуа) подставляется в выражение (11), все слагаемые, входящие в сумму в правой части формулы (11), обращаются в нуль в силу соотношения (8), кроме слагаемого, для которого $i=i\_{0},j=j\_{0}$ удовлетворяется. Отсюда следует, что

$F\left(x\_{i\_{0}},y\_{j\_{0}}\right)=f\left(x\_{i\_{0}},y\_{j\_{0}}\right)$ (12)

Мы видим, что многочлен (11) выполняет для многозначной логики те же функции, что и многочлен Жегалкина для двоичной логики, то есть соотношение (11) указывает на конкретную алгебраическую функцию, реализующую заданную таблицу истинности. Также видно, что соотношение (11) допускает обобщение на случай произвольного числа логических переменных.

Отметим, что методы управления на основе нечеткой логики в настоящее время активно развиваются [22,23]. Известны работы, в которых такие методы предлагается использовать для коррекции курса судов [24].

Очевидно, что если между лингвистическими переменными и элементами поля Галуа установлено соответствие один к одному, то все "команды" и "данные", преобразованные в такие переменные, могут быть в дальнейшем обработаны с помощью алгебраических функций, которые могут быть заведомо построены описанным выше способом.

Конечно, для реальных задач количество переменных, соответствующих 8-элементной розе компаса, недостаточно, но это не является препятствием.

Например, начиная с поля $GF(7)$, элементы которого можно выбрать как $(-3,-2,-1,0,1,2,3)$, мы можем построить поле $GF(7^{2})$.

Элементы этого поля также представимы в "двухкоординатной" форме (4), где коэффициенты $a\_{0}, a\_{1}$ принадлежат полю $GF(7)$.

Запись (4) в этом случае, для наглядности, можно интерпретировать, например, как дискретное представление вектора скорости (на плоскости), что полностью соответствует традиционному комплексному представлению векторов. Разница заключается в том, что при использовании поля $GF(7^{2})$компоненты скорости дискретны и могут быть отнесены к семи лингвистическим переменным "полный вперед, половина вперед, малый вперед, остановка двигателя, малый вперед, половина вперед, полный вперед".

Использование такого поля также позволяет нанести на карту лингвистические переменные, соответствующие 16-пунктовой розе компаса, рис. 2.



Рисунок 2 – 16-элементная роза компаса

А именно, количество ненулевых элементов поля $GF(7^{2})$ равно 48. Следовательно, все они являются корнями уравнения

$x^{48}-1=\left(x^{16}\right)^{3}-1=\left(x^{16}-1\right)\left(x^{32}+x^{16}+1\right)=0$(13)

Формула (13), помимо прочего, показывает, что среди элементов поля $GF(7^{2})$ существует 16 элементов, удовлетворяющих уравнению

$x^{16}-1=0$(14)

Эти 16 элементов можно рассматривать как корни 16-й степени из единицы, и они образуют группу при умножении. Следовательно, им можно присвоить лингвистические переменные, соответствующие 16-элементной розе компаса.

Таким образом, взаимно однозначное соответствие между многозначными $p^{n}$-логиками, где $p$ простое число, $n$ целое число и полями Галуа $GF(p^{n})$ создает все предпосылки для того, чтобы сделать операции над переменными многозначной логики как можно более понятными.

Можно утверждать, что визуализация в этом отношении обеспечивается не столько для переменных многозначной логики, сколько для элементов полей Галуа. Однако визуальное представление операций над переменными многозначных логик, отображаемых через поля Галуа, имеет и философский аспект, непосредственно связанный с проблемой интерпретации значений указанных переменных и с проблемой соотношения законов мышления и многозначных логик, затронутой в [8].

А именно, значение переменных бинарной логики связано с философской категорией истины. Эта категория относится к числу базовых понятий, вопрос о природе которых тесно связан с проблемой существования неопределенных понятий. Действительно, "определить" – значит раскрыть смысл одного термина через другие. Попытка раскрыть таким образом все имеющиеся в языке термины заведомо приводит к порочному кругу.

Объективная диалектика находит выход, определяя базовые категории через оппозиции "количество - качество", "содержание - форма" и т.д. Такой подход, в частности, был использован [13,25] для того, чтобы максимально корректно раскрыть смысл категории "информация", которую предлагалось рассматривать как философскую категорию в паре с категорией материи.

Проблема адекватной интерпретации понятия "информация", как подчеркивается в [26,27], становится все более актуальной в связи с исследованиями в области искусственного интеллекта, но для целей нашей статьи более важен сам подход "определение через противопоставление".

А именно, он показывает, что для определения базовых понятий наиболее важна структура связей между ними, а противопоставление - лишь одна из форм таких связей, причем та, которая заведомо соответствует двоичной логике и двоичным полям Галуа. Очевидно, что другие формы связей между базовыми понятиями не могут быть сведены к простому противопоставлению.

Это свидетельствует, например, о существовании ярко выраженного методологического (философского) аспекта разработки командных языков (даже на уровне конкретных технических систем), которые представляют собой замкнутое целое за счет отношений, записанных в алгебраической форме. Более того, разрабатывать замкнутые "языковые" системы на уровне абстракции крайне сложно. Гораздо удобнее (и нагляднее) делать это, решая конкретные задачи, например, связанные с управлением движущимся транспортом, в терминах нечеткой логики, преобразованной в алгебраическую форму.

Такая постановка вопроса делает еще более актуальной задачу обеспечения наглядности и удобства использования многозначных логик. В следующем разделе рассматриваются конкретные вычислительные инструменты, ориентированные на использование логик, соответствующих полям $GF(7^{n})$.

Этот пример позволяет наглядно продемонстрировать, что можно реализовать различного рода устройства, выполняющие вычисления в терминах многозначной логики, но при этом построенные на базе типовых электронных компонентов, использующих двоичную логику.

**1.5 Вычислительная реализация семизначных логических операций**

В настоящее время алгоритмы и схемы радиоэлектронных устройств, выполняющих вычисления по модулю, широко представлены в литературе. Так, подобные алгоритмы используются в устройствах шифрования, кодирования, при сжатии и передаче информации, в устройствах автоматики [28-30].

Как было показано выше, любые функции, аргументами которых являются переменные, принимающие значения в поле Галуа, могут быть явно сведены к алгебраическим выражениям, включающим только операции умножения и сложения по модулю $p$.

Следовательно, умножители и сумматоры по модулю $p$ являются основой для автоматизации любых операций над логическими (лингвистическими) переменными. Устройства такого типа могут быть реализованы довольно простыми средствами, как будет показано ниже.

Блок-схема умножителя рассматриваемого типа представлена на рис. 3. В состав схемы входят сумматоры (маркировка на схеме $Σ$), которые подсчитывают количество единиц на входах $a\_{i}$ соответствующих представлению числа в двоичной форме. Предполагается, что на вход системы не поступают сигналы, соответствующие числу 7 или 0. Это допустимо, так как при вычислении по модулю 7 $7≡0(7)$ имеет место, следовательно, в этом случае вычисленное произведение равно нулю. В этом случае $Σa\_{i}$ может принимать значения 1 или 2, так как в двоичной системе счисления чисел, изменяющихся от 1 до 6, существует не менее одной и не более двух единиц.

Тогда

$6∙a\_{3}a\_{2}a\_{1}=\_{(7)}\overbar{a}\_{3}\overbar{a}\_{2}\overbar{a}\_{1}$(15)

где $a\_{i}$ символы в двоичной системе счисления, штрих над символом означает операцию инверсии, т.е. 0 меняется на 1 и наоборот.

В силу ассоциативности умножения по модулю, произведение любых двух ненулевых элементов поля $GF(7)$ можно свести к умножению двух чисел в двоичном представлении, причем в обоих этих числах только один из символов $a\_{i}$ будет ненулевым.



Рисунок 3 – Блок-схема умножения модуля на семь

Соответствующую работу реализует блок инверторов (в схеме используется стандартное обозначение инвертора), управляемый сигналом, поступающим с $Σ$ элементов. Если на выходе этих элементов формируется логический ноль, то сигналы $a\_{i}$ и $b\_{i}$ остаются неизменными, если логическая единица - принимают обратные значения.

Сигнал устанавливает $\tilde{a}\_{i}$ и $\tilde{b}\_{i}$приведенные к формату, в котором только одна из переменных этих наборов ненулевая, подаются на блок прямого умножения (схемное обозначение -⊗ ).

Установленный сигнал $\tilde{c}\_{i}$ с выхода прямого умножителя подается на блок выходного инвертора, который работает так же, как и блок входного инвертора.

Принципиальная схема блока прямого умножения показана на рис. 4.

Этот блок работает следующим образом.

Простое число 7 является частным случаем простых чисел Мерсенна, представленных в виде $p\_{m}=2^{n}-1$. Такие числа обладают следующим свойством. При умножении любого числа на 2 по модулю $p\_{m}$ приводит к циклической перестановке символов. Например,

$2∙a\_{2}a\_{1}a\_{0}=\_{(7)}a\_{1}a\_{0}a\_{2}$ (16)

где $a\_{i}$ двоичные символы.



Рисунок 4 – Схема прямого умножителя

Рассмотрим произведение двух чисел $B∙A$ записанных в двоичной системе счисления. У нас есть

$B∙A=\_{\left(2\right)}b\_{3}∙2^{2}∙A+b\_{2}∙2^{1}∙A∙+b\_{1}∙2^{0}∙A$ (17)

Согласно формуле (16), произведения $2^{m}∙A$ могут быть записаны через циклические перестановки, т.е. произведение, вычисляемое по модулю 7, является суммой следующих трех чисел, записанных в двоичной форме как

$b\_{1}∙a\_{3}a\_{2}a\_{1}$ (18)

$b\_{2}∙a\_{2}a\_{1}a\_{3}$ (19)

$b\_{3}∙a\_{1}a\_{3}a\_{2}$ (20)

где только одно из $b\_{i}$ значений равна 1, а остальные равны 0.

Каждое из двоичных трехзначных чисел, фигурирующих в формулах (18) - (20), также можно записать в виде степени двойки.

Следовательно, результат умножения в вычислениях по модулю 7 можно записать как

$B∙A=\_{\left(2\right)}c\_{3}∙2^{2}+c\_{2}∙2^{1}∙+c\_{1}∙2^{0}$, (21)

где

$c\_{3}=b\_{1}a\_{3}+b\_{2}a\_{2}+b\_{3}a\_{1}$ (22)

$c\_{2}=b\_{1}a\_{2}+b\_{2}a\_{1}+b\_{3}a\_{3}$ (23)

$c\_{1}=b\_{1}a\_{1}+b\_{2}a\_{3}+b\_{3}a\_{2}$ (24)

Поскольку в рассматриваемой схеме используются инверторы, то в формулах (22) - (24) только одна из величин $a\_{i}$ и только одно из значений $b\_{i}$ равна 1, остальные равны 0. Следовательно, среди всех значений $c\_{i}$ только одно также равно 1, а остальные равны 0.

Таким образом, результат произведения соответствует трем выходам схемы, на которых формируются логические переменные $c\_{i}$ формируются.

Поскольку из всех значений $b\_{i}$ только одно равно 1, то возможны три варианта

Если $b\_{1}=1$то

$\left(c\_{3},c\_{2},c\_{1}\right)=\left(a\_{3},a\_{2},a\_{1}\right)$ (25)

Если $b\_{2}=1$то

$\left(c\_{3},c\_{2},c\_{1}\right)=\left(a\_{2},a\_{1},a\_{3}\right)$ (26)

Если $b\_{3}=1$то

$\left(c\_{3},c\_{2},c\_{1}\right)=\left(a\_{1},a\_{3},a\_{2}\right)$ (27)

Если $b\_{1}=1$ то состояние выходов ci повторяет состояние входов $a\_{i}$, если $b\_{2}=1$, то существует циклическая перестановка на одну позицию вправо, а если $b\_{3}=1$, то на одну позицию влево.

Схема, обеспечивающая такую перестановку, может быть реализована различными способами. Один из них основан на наборе операций, которые схематично можно представить следующим образом.

$NO \left[\left(0,1,0\right) OR \left(0,0,1\right)\right]\rightarrow NO \left(0,1,1\right)\rightarrow \left(1,0,0\right)$ (28)

$\left(0,1,0\right) OR \left(0,0,0\right)\rightarrow \left(0,1,0\right)$ (29)

Эти обозначения подразумевают, что $NO$ и $OR$ применяются к каждой из булевых переменных, встречающихся в последовательностях. Формулы (28) и (29) показывают лишь частный случай; очевидно, что они остаются справедливыми и для циклических перестановок.

Из этих формул следует, что перестановка, соответствующая формулам (25) - (27), также может быть реализована способом, который реализуется схемой рис. 4.

В этом случае, если $b\_{1}=0$то выполняется дополнительное инвертирование сигнала, что соответствует выполнению операции (28). В соответствии со схемой на рис. 4 эту операцию выполняет сумматор, выход которого подключен к выходному инвертору.

Полная схема, выполненная в приложении NI Multisim [31], показана на рис. 5.



Рисунок 5 – Схема умножителя по модулю семь, собранная в программе NI Multisim

Как следует из приведенного выше описания множителя, его схема учитывает важнейшие специфические особенности вычислительных систем, выполняющих операции в полях Галуа, которые непосредственно связаны с операциями многозначной логики.

**1.6 Заключение первой статьи**

В данной работе показано, что проблема интерпретации переменных многозначных логик не обязательно должна решаться с привлечением философской категории истины. Возможным вариантом является использование тесной связи переменных многозначных логик, число элементов которых равно степени простого числа, с полями Галуа. В этом случае можно, в частности, установить связь между переменными многозначной логики и лингвистическими переменными, используемыми в нечеткой логике. Кроме того, эта связь позволяет свести любые операции над переменными многозначной логики к вычислению алгебраических функций, аргументы которых принимают значение в поле Галуа. Иначе, любые операции многозначных логик указанного типа можно свести к операциям сложения и умножения по модулю степени простого числа.

Такие операции, в свою очередь, могут быть реализованы с помощью радиоэлектронных схем, собранных на типовых элементах, выполняющих операции двоичной логики. При этом, как показано в примере реализации таких схем, они могут быть достаточно простыми.

**1.7 Список использованных источников первой статьи**

1. Lukasiewicz J. On three-valued logic // The Polish Review. – 1968. – С. 43-44.
2. Gomes E.L. Thinking about Contradictions: The Imaginary Logic of Nikolai Aleksandrovich Vasil’ev: V. Raspa, Translated by Peter N. Dale. – Heidelberg, New York: Springer International Publishing AG, 2017. – XXI+ 160 pp. – Hardcover US 109.99, e-book US 84.99. – Hardcover ISBN 978-3-319-66085-1. – eBook ISBN 978-3-319-66086-8. – URL: https://doi.org/10.1080/01445340.2019.1591669.
3. Kline M. Mathematics: The loss of certainty. – Galaxy Books, 1982. – Vol. 686.
4. Jo S.B., Kang J., Cho J.H. Recent advances on multivalued logic gates: a materials perspective // Advanced Science. – 2021. – Vol. 8(8). – URL: https://doi.org/10.1002/advs.202004216.
5. Yoo H., Kim C.H. Multi-valued logic system: New opportunities from emerging materials and devices // Journal of Materials Chemistry C. – 2021. – Vol. 9(12). – P. 4092-4104. – DOI: https://doi.org/10.1039/D1TC00148E.
6. Zamansky A. On recent applications of paraconsistent logic: an exploratory literature review // Journal of Applied Non-Classical Logics. – 2019. – Vol. 29(4). – P. 382-391. – URL: https://doi.org/10.1080/11663081.2019.1656393.
7. Hernández-Tello A., Macías V.B., Coniglio M.E. Paracomplete Logics Dual to the Genuine Paraconsistent Logics: The Three-valued Case // Electronic Notes in Theoretical Computer Science. – 2020. – Vol. 354. – P. 61-74. – URL: https://doi.org/10.1016/j.entcs.2020.10.006.
8. Gabrielyan O., Vitulyova E., Suleimenov I. Multi-Valued Logics as an Advanced Basis for Artificial Intelligence (As an Example of Applied Philosophy) // Wisdom. – 2022. – Vol. 21(1). – P. 170-181. – URL: https://doi.org/10.24231/wisdom.v21i1.721.
9. Nakayama Y., Akama S., Murai T. Deduction System for Decision Logic Based on Many-valued Logics // International Journal on Advances in Intelligent Systems. – 2018. – Vol. 11(1/2). – P. 115-126.
10. Lethen T. Gödel on many-valued logic // The Review of Symbolic Logic. – 2021. – P. 1-17. – URL: https://doi.org/10.1017/S1755020321000034.
11. Rescher N. Topics in philosophical logic. – Springer Science & Business Media, 2013. – Vol. 17.
12. Kahane H., Hausman A., Boardman F. Logic and philosophy: A modern introduction. – Hackett Publishing, 2020.
13. Suleimenov I.E., Gabrielyan O.A., Bakirov A.S., Vitulyova Y.S. Dialectical understanding of information in the context of the artificial intelligence problems // IOP Conference Series: Materials Science and Engineering. – 2019. – Vol. 630(1). – P. 012007. – IOP Publishing.
14. Suleimenov I., Bakirov A., Moldakhan I. Formalization of ternary logic for application to digital signal processing // Energy Management of Municipal Transportation Facilities and Transport. – 2019. – P. 26-35. – Springer, Cham. – URL: https://doi.org/10.1007/978-3-030-57453-6\_3.
15. Dey S., Ghosh R. 4-bit crypto S-boxes: Generation with irreducible polynomials over Galois field GF (24) and cryptanalysis // Cryptology ePrint Archive. – 2018.
16. Wang N., Di G., Lv X., Hou M., Liu D., Zhang J., Duan X. Galois field-based image encryption for remote transmission of tumor ultrasound images // IEEE Access. – 2019. – Vol. 7. – P. 49945-49950. – DOI: 10.1109/ACCESS.2019.2910563.
17. Khari M., Garg A.K., Gandomi A.H., Gupta R., Patan R., Balusamy B. Securing data in Internet of Things (IoT) using cryptography and steganography techniques // IEEE Transactions on Systems, Man, and Cybernetics: Systems. – 2019. – Vol. 50(1). – P. 73-80. – DOI: 10.1109/TSMC.2019.2903785.
18. Moldakhan I., Matrassulova D.K., Shaltykova D.B., Suleimenov I.E. Some advantages of non-binary Galois fields for digital signal processing // Indonesian Journal of Electrical Engineering and Computer Science. – 2021. – Vol. 23(2). – P. 871-877. – DOI: 10.11591/ijeecs.v23.i2.pp871-878 .
19. Vitulyova E.S., Matrassulova D.K., Suleimenov I.E. New application of non-binary Galois fields Fourier transform: Digital analog of convolution theorem // Indonesian Journal of Electrical Engineering and Computer Science. – 2021. – Vol. 23(3). – P. 1718-1726. – DOI: 10.11591/ijeecs.v23.i3.pp1718-1726 .
20. Mittal K., Jain A., Vaisla K.S., Castillo O., Kacprzyk J. A comprehensive review on type 2 fuzzy logic applications: Past, present and future // Engineering Applications of Artificial Intelligence. – 2020. – Vol. 95. – P. 103916. – URL: https://doi.org/10.1016/j.engappai.2020.103916.
21. Marcos J. On a problem of da Costa // Essays on the Foundations of Mathematics and Logic. – 2005. – Vol. 2. – P. 39-55.
22. Rezk H., Aly M., Al-Dhaifallah M., Shoyama M. Design and hardware implementation of new adaptive fuzzy logic-based MPPT control method for photovoltaic applications // IEEE Access. – 2019. – Vol. 7. – P. 106427-106438. – DOI: 10.1109/ACCESS.2019.2932694.
23. Sharma R., Bhasin S., Gaur P., Joshi D. A switching-based collaborative fractional order fuzzy logic controllers for robotic manipulators // Applied Mathematical Modelling. – 2019. – Vol. 73. – P. 228-246. – URL: https://doi.org/10.1016/j.apm.2019.03.041.
24. Sedova N., Sedov V., Bazhenov R., Bogatenkov S. Neural network classifier for automatic course-keeping based on fuzzy logic // Journal of Intelligent & Fuzzy Systems. – 2021. – Vol. 40(3). – P. 4683-4694. – DOI: 10.3233/JIFS-201495.
25. Vitulyova Y. S., Bakirov A. S., Baipakbayeva S. T., Suleimenov I. E. Interpretation of the category of “complex” in terms of dialectical positivism // IOP Conference Series: Materials Science and Engineering. – 2020. – Vol. 946, No. 1. – P. 012004. – DOI: 10.1088/1757-899X/946/1/012004.
26. Kalimoldayev M. N., Pak I. T., Baipakbayeva S. T., Mun G. A., Shaltykova D. B., Suleimenov I. E. Methodological basis for the development strategy of artificial intelligence systems in the Republic of Kazakhstan in the message of the president of the Republic of Kazakhstan dated October 5, 2018 // News of the National Academy of Sciences of the Republic of the Kazakhstan – Series of geology and technical sciences. – 2018. – Vol. 6. – P. 47-54. – URL: https://doi.org/10.32014.2018.2518-170X.34.
27. Suleimenov I. E., Vitulyova Y. S., Bakirov A. S., Gabrielyan O. A. Artificial Intelligence: what is it? // In Proceedings of the 2020 6th International Conference on Computer and Technology Applications. – 2020. – P. 22-25. – URL: https://doi.org/10.1145/3397125.3397141.
28. Moysis L., Petavratzis E., Volos C., Nistazakis H., Stouboulos I. A chaotic path planning generator based on logistic map and modulo tactics // Robotics and Autonomous Systems. – 2020. – Vol. 124. – P. 103377. – URL: https://doi.org/10.1016/j.robot.2019.103377.
29. Prasanna D., Sriram C., Murthy C. R. On the identifiability of sparse vectors from modulo compressed sensing measurements // IEEE Signal Processing Letters. – 2020. – Vol. 28. – P. 131-134. – DOI: 10.1109/LSP.2020.3047584.
30. Ţiplea F. L., Iftene S., Teşeleanu G., Nica A. M. On the distribution of quadratic residues and non-residues modulo composite integers and applications to cryptography // Applied Mathematics and Computation. – 2020. – Vol. 372. – P. 124993. – URL: https://doi.org/10.1016/j.amc.2019.124993.
31. NI Multisim 14.1 – URL: <https://www.ni.com/ru-ru/support/downloads/software-products/download.multisim.html#306441> т(дата обращения: 18.11.2024).

**2 Особенности алгоритмов цифровой обработки сигналов с использованием полей Галуа GF(2n+1)**

**2.1 Аннотация** **второй статьи**

Предложено знакопеременное представление целых чисел в двоичной форме, в котором вместо нулей и единиц используются числа -1 и +1. Показано, что такое представление создает значительные удобства для умножения чисел по модулю $p=2^{n}+1$. Для таких чисел можно реализовать алгоритм умножения по модулю p, аналогичный алгоритму умножения по модулю числа Мерсенна. Показано, что для таких чисел может быть предложен простой алгоритм вычисления цифрового логарифма. Этот алгоритм позволяет, в частности, свести операцию умножения по модулю простого числа $p=2^{n}+1 $к операции сложения.

**2.2 Введение** **второй статьи**

В настоящее время небинарные поля Галуа находят все большее применение в информационных технологиях [1,2], в частности, в системах информационной безопасности [3,4]. В качестве примера можно привести многообразие полей Галуа $GF\left(p\right)$которые являются классами остатков кольца целых чисел по модулю некоторого простого числа $p$. Преимущества использования таких полей для цифровой обработки сигналов были наглядно продемонстрированы в работах [5,6].

В современной литературе также имеется ряд докладов, посвященных разработке электронных схем, выполняющих операции сложения и умножения по модулю, например, [7,8]. Подобные доклады тесно связаны с исследованиями в области практического использования полей Галуа, поскольку операции сложения и умножения по модулю целого числа можно рассматривать как операции над элементами поля Галуа. Этот интерес связан, в частности, с тем, что такие операции представляют значительный интерес для криптографии [9, 10].

Разработка электронных схем, работающих в полях Галуа, представляет интерес и с точки зрения совершенствования систем искусственного интеллекта (ИИ). А именно, как было показано в докладах [11, 12], дальнейшее развитие ИИ, предполагающее его постепенное приближение к биологическому прототипу, не может исключать перехода к многозначной логике, поскольку человеческое мышление несводимо к двоичной логике. Уместно подчеркнуть, что проблема создания ИИ, приближающегося к биологическому прототипу, представляет значительный интерес, в том числе с точки зрения раскрытия сущности интеллекта как такового [13,14].

Многозначная логика, восходящая к работам Яна Лукасевича [15], активно развивается в последнее время [16-19], однако уровень ее практического использования явно не соответствует имеющемуся потенциалу [11]. Если число значений, принимаемых переменными многозначной логики, равно целой степени некоторого простого числа, то этим значениям можно поставить в соответствие один к одному элементы поля Галуа. Следовательно, любые операции в этом случае сводятся к сложению и умножению, т.е. совершенствование электронных схем, выполняющих сложение и умножение по модулю целого числа, представляет интерес и с этой точки зрения. Возникающие при этом возможности раскрыты, в частности, в [20].

В свою очередь, среди простых полей Галуа $GF\left(p\right), $особое место занимают поля, для которых число $p$ равно простому числу Мерсенна. Такие числа можно представить в виде

$p=2^{n}-1$ (1)

где $n$ специально выбираемые целые числа, первыми из которых являются 2, 3, 5 и 7.

Такие числа используются, в частности, для генерации псевдослучайных чисел [19]. В частности, известен генератор псевдослучайных чисел Mersenne twister, разработанный в 1997 году японскими учеными М. Мацумото и Т. Нисимурой [21], непосредственно основанный на использовании чисел Мерсенна.

Существуют работы, в которых обосновывается целесообразность использования таких чисел для передачи данных [21,22] и др.

Очень примечательной особенностью чисел Мерсенна $p\_{m}$ является тот факт, что поля Галуа $GF\left(p\_{m}\right)$ позволяет реализовать достаточно простые электронные схемы, выполняющие операции сложения и умножения по модулю $p\_{m}$т.е. те операции, к которым сводятся любые другие операции над элементами поля $GF\left(p\_{m}\right)$ сводятся. Примеры таких систем и алгоритмы их функционирования отражены в современной литературе, например [23].

В данной работе показано, что поля Галуа, $GF\left(p\right),$, для которых $p=2^{n}+1$., также представляют значительный интерес для цифровой обработки сигналов. Одним из таких полей является поле $GF\left(257\right)$ поле, которое целесообразно использовать для цифровой обработки сигналов со стандартным числом уровней, равным 256. Таким образом, можно утверждать, что для данного случая существует вполне конкретное поле Галуа, которое, помимо всего прочего, позволяет приводить сигналы, удовлетворяющие существующим стандартам [24], к логическим операциям.

**2.3 Преимущества чисел Мерсенна для вычисления систем по модулю целого числа**

В связи с цифровой обработкой сигналов представляет интерес следующее свойство чисел Мерсенна. При умножении числа $a$ записанного в двоичной форме, на 2 по модулю числа Мерсенна повторяется до циклической перестановки символов. Например, для вычислений в поле $GF\left(7\right)$справедливо следующее равенство

$2∙a\_{2}a\_{1}a\_{0}=\_{(7)}a\_{1}a\_{0}a\_{2}$ (2)

где $a\_{i}$ двоичные символы.

Удобство использования этого свойства для цифровой обработки сигналов в полях Галуа заключается в следующем. Рассмотрим поле $GF\left(127\right)$ характеристикой которого является число Мерсенна с $n=7$. Отметим, что этот пример важен и с практической точки зрения, поскольку шкала со 127 уровнями часто используется и в цифровой обработке сигналов. В двоичном виде любой из элементов поля $GF\left(127\right) $поля может быть представлен как

$A=2^{6}∙a\_{6}+2^{5}∙a\_{5}+…+2^{1}∙a\_{1}+2^{0}∙a\_{0}$ (3)

Подчеркнем, что количество элементов, представленных в форме (3), при вычислении в терминах обычных целых чисел равно 128, но

$1111111≡0000000\left(127\right)$ (4)

Следовательно, число элементов поля, заданное формулой (3), действительно равно 127.

Пусть элемент $A$представленный в форме (3), умножается по модулю 127 на элемент $B$представленный в той же форме.

$B=2^{6}∙b\_{6}+2^{5}∙b\_{5}+…+2^{1}∙b\_{1}+2^{0}∙b\_{0}$ (5)

Тогда результатом произведения будет сумма следующих членов

$$A∙b\_{0}=\left(2^{6}∙a\_{6}+2^{5}∙a\_{5}+…+2^{1}∙a\_{1}+2^{0}∙a\_{0}\right)b\_{0}$$

$2^{1}∙A∙b\_{1}=\left(2^{6}∙a\_{5}+…+2^{2}∙a\_{1}+2^{1}∙a\_{0}+2^{0}∙a\_{6}\right)b\_{1}$ (6)

...

$$2^{6}∙A∙b\_{6}=\left(2^{6}∙a\_{0}+2^{5}∙a\_{6}+…+2^{1}∙a\_{2}+2^{0}∙a\_{1}\right)b\_{6}$$

Группируя члены с одинаковыми степенями двойки, получаем

$$c\_{6}=a\_{6}b\_{0}+a\_{5}b\_{1}+…+a\_{0}b\_{6}$$

 $c\_{5}=a\_{5}b\_{0}+a\_{4}b\_{1}+…+a\_{6}b\_{6}$ (7)

...

$$c\_{0}=a\_{0}b\_{0}+a\_{6}b\_{1}+…+a\_{1}b\_{6}$$

Подчеркнем, что хотя в формулы (7) входят только величины, принимающие значения 0 или 1, на сам результат суммирования такие ограничения не накладываются, т.е. сумма вычисляется в смысле исходного поля $GF\left(127\right)$.

Алгоритм, основанный на формулах (7), допускает достаточно простую схемную реализацию, поэтому имеет смысл рассмотреть, нет ли возможности реализовать его аналог для полей $GF\left(2^{n}+1\right)$в частности, для поля $GF\left(257\right)$. Это связано с очевидным соображением: шкала оцифровки сигнала, предусматривающая использование 256 уровней, является одной из наиболее распространенных [24].

**2.4 Алгоритмы вычислений по модулю в частном случае поля** $GF\left(17\right)$

Начнем с рассмотрения одного из простейших полей типа $GF\left(2^{n}+1\right)$, а именно с поля $GF\left(17\right)$.

Это поле, как и другие поля Галуа, представляет собой кольцо классов остатков кольца целых чисел, в данном случае по модулю 17. Традиционно при представлении элементов поля используются положительные целые числа, однако это вовсе не обязательно. В частности, как подчеркивается в [25], для представления элементов поля целесообразно использовать набор элементов $\left\{-1,0,1\right\}$ для представления поля $GF\left(3\right)$где использование фигурных скобок подчеркивает, что рассматривается множество.

Аналогично, поле $GF\left(17\right)$ можно рассматривать как набор элементов

$GF\left(17\right)=\left\{-8,-7,…,0,…7,8\right\}$ (8)

Выбор представляющих элементов может быть произвольным, например, вплоть до операции сравнения по модулю,

$-8≡9(17)$ (9)

Преимущества такого выбора для целей данной работы демонстрируют таблицы 1 и 2. В этих таблицах степени элементов рассматриваемого поля считаются в обычном представлении через целые положительные числа и в представлении (8), соответственно.

Обе таблицы демонстрируют тот факт, что все ненулевые элементы рассматриваемого поля, как следует из общей теории полей Галуа, подчиняются уравнению

$x^{16}-1=0$ (10)

Таблица 1. Степени элементов $GF\left(17\right)$ поля в терминах целых положительных чисел

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| $$n=0$$ | $$n=2$$ | $$n=4$$ | $$n=8$$ | $$n=16$$ |
| 1 | 1 | 1 | 1 | 1 |
| 2 | 4 | 16 | 1 | 1 |
| 3 | 9 | 13 | 16 | 1 |
| 4 | 16 | 1 | 1 | 1 |
| 5 | 8 | 13 | 16 | 1 |
| 6 | 2 | 4 | 16 | 1 |
| 7 | 15 | 4 | 16 | 1 |
| 8 | 13 | 16 | 1 | 1 |
| 9 | 13 | 16 | 1 | 1 |
| 10 | 15 | 4 | 16 | 1 |
| 11 | 2 | 4 | 16 | 1 |
| 12 | 8 | 13 | 16 | 1 |
| 13 | 16 | 1 | 1 | 1 |
| 14 | 9 | 13 | 16 | 1 |
| 15 | 4 | 16 | 1 | 1 |
| 16 | 1 | 1 | 1 | 1 |

Из этого соотношения, в частности, следует, что любой элемент рассматриваемого поля может быть представлен в виде

$x=\left(\sqrt[2]{1}\right)\_{4}^{s\_{4}}\left(\sqrt[4]{1}\right)\_{3}^{s\_{3}}\left(\sqrt[8]{1}\right)\_{2}^{s\_{2}}\left(\sqrt[16]{1}\right)\_{1}^{s\_{1}}$ (11)

где $s\_{i}=0;1$.

Мы покажем это, раскрыв сначала смысл формальных обозначений $\left(\sqrt[k\_{2}]{1}\right)\_{k\_{1}}$используя табл. 1 и 2.

Соотношение (10) можно переписать в виде

$\left(x^{8}\right)^{2}-1=0$ (12)

Формула (12) подчеркивает, что существует только два различных значения элемента $z=x^{8}$. Это соответствует тому, что в четвертом столбце таблиц 1 и 2 есть только два различных элемента.

Есть еще две похожие формы представления отношения (10).

$\left(x^{4}\right)^{4}-1=0$ (13)

$\left(x^{2}\right)^{8}-1=0$ (14)

Формула (13) подчеркивает, что существует четыре различных значения элемента $z=x^{4}$которые приведены в третьих, колонках таблиц 1 и 2. Аналогично, как показывает соотношение (14), существует восемь различных элементов $z=x^{2}$, которые находятся во вторых столбцах этих таблиц.

Таблица 2. Степени элементов поля $GF\left(17\right)$ в представлении (8)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| $$n=0$$ | $$n=2$$ | $$n=4$$ | $$n=8$$ | $$n=16$$ |
| 1 | 1 | 1 | 1 | 1 |
| 2 | 4 | -1 | 1 | 1 |
| 3 | -8 | -4 | -1 | 1 |
| 4 | -1 | 1 | 1 | 1 |
| 5 | 8 | -4 | -1 | 1 |
| 6 | 2 | 4 | -1 | 1 |
| 7 | -2 | 4 | -1 | 1 |
| 8 | -4 | -1 | 1 | 1 |
| -8 | -4 | -1 | 1 | 1 |
| -7 | -2 | 4 | -1 | 1 |
| -6 | 2 | 4 | -1 | 1 |
| -5 | 8 | -4 | -1 | 1 |
| -4 | -1 | 1 | 1 | 1 |
| -3 | -8 | -4 | -1 | 1 |
| -2 | 4 | -1 | 1 | 1 |
| -1 | 1 | 1 | 1 | 1 |

Соответственно, формально мы можем записать:

$\left(\sqrt[16]{1}\right)\_{1}=16$ (15)

$\left(\sqrt[16]{1}\right)\_{2}=4;13$ (16)

$\left(\sqrt[16]{1}\right)\_{3}=2;8;9;15$ (17)

Восемь оставшихся элементов y, выделенных в таблицах 1 и 2 цветом, представляют собой примитивные элементы. Они обладают тем свойством, что $y^{m}=1$ если и только если $m=16$. Мы имеем

$\left(\sqrt[16]{1}\right)\_{4}=3;5;6;7;10;11;12;14$ (18)

Доказанное соотношение (11) следует непосредственно из формул (15) - (18), а выбор элементов $\left(\sqrt[k\_{2}]{1}\right)\_{k\_{1}}$определяется следующими соображениями. Любой ненулевой элемент рассматриваемого поля представляет собой некоторую степень одного из примитивов $y$ перечисленных в правой части формулы (18). Это следует из общей теории полей Галуа, и для наглядности это можно показать следующим образом.

Все степени примитивного элемента y рассматриваемого поля от 0 до 15 различны. В то же время все эти степени являются корнями уравнения (10), то есть исчерпывают элементы поля.

Рассмотрим мощность $y^{m}$ и представим число m, где $0\leq m\leq 15$ в двоичной форме

$m=m\_{4}m\_{3}m\_{2}m\_{1}$ (19)

где $m\_{i}$ двоичные символы.

Такая запись означает, что

$m=2^{3}∙m\_{4}+2^{2}∙m\_{3}+2^{1}∙m\_{2}+2^{0}∙m\_{1}$ (20)

Таким образом, степень $y^{m}$ можно представить в виде

$y^{m}=\left(y^{8}\right)^{m\_{4}}\left(y^{4}\right)^{m\_{3}}\left(y^{2}\right)^{m\_{2}}\left(y^{1}\right)^{m\_{1}}$ (21)

Это выражение совпадает с (11), если положить соответствующий корень единственности равным одной из мощностей примитивного элемента, фигурирующего в (21).

Для дальнейшего существенно, что при переходе к представлению элементов рассматриваемого поля в виде (8), корни единственности (15) - (18) приобретают симметричный вид, что и показывает табл. 2.

$\left(\sqrt[16]{1}\right)\_{1}=-1$ (22)

$\left(\sqrt[16]{1}\right)\_{2}=4;-4$ (23)

$\left(\sqrt[16]{1}\right)\_{3}=2;8;-8;-2$ (24)

$\left(\sqrt[16]{1}\right)\_{4}=3;5;6;7;-7;-6;-5;-3$ (25)

Это определяет удобство представления в виде (8): все элементы рассматриваемого поля, за исключением примитивных, представимы в виде степени двойки, что также подчеркивает табл. 2.

Это обстоятельство порождает еще один эффективный алгоритм умножения элементов поля, де-факто основанный на методе цифрового логарифма. Цифровой логарифм рассматривался во многих работах, в частности в [26,27], но эта проблема, если ее поставить в общем виде, остается нерешенной.

Однако для практических нужд можно ограничиться ее решением для конкретных полей Галуа. В частности, в рамках данной работы она решается применительно к полям Галуа вида $GF\left(2^{p}+1\right)$, которое, как отмечалось выше, включает в себя также поле $GF\left(257\right)$, что соответствует часто используемому на практике числу уровней цифрового сигнала.

По отношению к $GF\left(17\right) $поле, алгоритм цифрового логарифма может быть построен следующим образом.

1. Элемент поля идентифицируется как принадлежащий к набору примитивных элементов.
2. Если данный элемент принадлежит указанному множеству, то значение $m\_{1}$ в формуле (21) выбирается равным 1, если нет, то нулю.
3. Если $m\_{1}=1$то рассматриваемый элемент умножается на 3 по модулю 17. В результате представления в виде (8) элемент сокращается до степени двойки. При использовании двоичной системы счисления это означает, что логическая единица будет стоять только на одной из позиций, определяющих экспоненты $m\_{2}$, $m\_{3}$ и $m\_{4}$.
4. Набор экспонент степеней $m\_{i} $завершает процедуру цифрового логарифма.

Решение задачи цифрового логарифма, в свою очередь, позволяет существенно упростить алгоритм умножения элементов поля друг на друга. Действительно, в этом случае операция сводится к сложению чисел (20) по модулю 16. Выполнение такой операции схемными средствами не вызывает затруднений, поскольку сводится к обычному сложению двоичных чисел с отбрасыванием старшего разряда.

Описанный алгоритм действительно позволяет существенно упростить операцию умножения в $GF\left(17\right)$ поле, но, во-первых, вопрос схемной идентификации примитивных элементов остается открытым, а во-вторых, этот алгоритм очень специфичен. Действительно, он построен на том, что любой из примитивных элементов сокращается до степени двойки (с положительным или отрицательным знаком) путем умножения на фиксированный элемент (например, на 3). Эта ситуация не реализуется для других полей Галуа рассматриваемого типа, в частности, для поля $GF\left(257\right)$которое представляет основной интерес с точки зрения практических приложений.

**2.5 Попеременное двоичное кодирование**

Мы показываем, что указанная проблема решается с помощью поочередного кодирования элементов полей Галуа рассматриваемого типа.

Опять же, давайте начнем с примера с полем $GF\left(17\right)$. Рассмотрим выражение вида (3), но только теперь будем понимать под $a\_{i}$ числа, принимающие значения +1 и -1.

$A=2^{3}∙a\_{3}+2^{2}∙a\_{2}+2^{1}∙a\_{1}+2^{0}∙a\_{0}$ (26)

Легко показать, что результат вычислений по формуле (3) в этом случае обязательно даст нечетное число.

Всего существует $2^{4}$ таких комбинаций вида (3), а максимальное число составляет

$A\_{max}=2^{3}+2^{2}+2^{1}+2^{0}=15$, (27)

и, соответственно, $A\_{min}=-15$.

В противном случае множество чисел, задаваемых формулой (26), совпадает с множеством нечетных чисел в диапазоне от -16 до +16. Нуль мы исключаем из рассмотрения, что вполне оправдано, поскольку рассматривается операция умножения. Тогда количество комбинаций вида (26) в рассматриваемом случае совпадает с количеством ненулевых элементов поля $GF\left(17\right)$.

Все возможные комбинации перечислены в таблицах 3 и 4. Таблица 3 относится к элементам поля, которые не являются примитивными, таблица 4 - наоборот. Первые четыре столбца этих таблиц отображают числа $a\_{i}$ встречающиеся в выражениях (26), в пятом столбце - результат суммирования по обычным целым числам $A\_{1}$а в шестой - результат приведения по модулю 17 к виду (8) A.

Таблица 3. Попеременное двоичное представление элементов $GF\left(17\right)$ которые не являются примитивами

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| $$a\_{3}$$ | $$a\_{2}$$ | $$a\_{1}$$ | $$a\_{0}$$ | $$A\_{1}$$ | $$A$$ |
| 1 | -1 | -1 | -1 | 1 | 1 |
| -1 | -1 | -1 | -1 | -15 | 2 |
| -1 | -1 | -1 | 1 | -13 | 4 |
| -1 | -1 | 1 | 1 | -9 | 8 |
| -1 | 1 | 1 | 1 | -1 | 1 |
| 1 | 1 | 1 | 1 | 15 | -2 |
| 1 | 1 | 1 | -1 | 13 | -4 |
| 1 | 1 | -1 | -1 | 9 | -8 |
| 1 | -1 | -1 | -1 | 1 | 1 |

Видно, что поле модульной редукции чисел, представленных в виде (26), исчерпывает множество ненулевых элементов поля $GF\left(17\right)$. Следовательно, такое представление может быть использовано наряду с любым другим, особенно если учесть, что представители классов остатков кольца целых чисел по модулю простого числа могут быть выбраны произвольно.

Применительно к рассматриваемому полю представление вида (26), в котором $a\_{i}=\pm 1$ обладает свойством, аналогичным тому, которым обладают простые числа Мерсенна. А именно, умножение числа, записанного в представлении (26), которое далее мы будем называть знакопеременным, на два можно отобразить в виде следующей операции

$2∙A=2^{3}∙a\_{2}+2^{2}∙a\_{1}+2^{1}∙a\_{0}-2^{0}∙a\_{3}$ (28)

Таблица 4. Попеременное двоичное представление примитивных элементов поля $GF\left(17\right)$ поля

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| $$a\_{3}$$ | $$a\_{2}$$ | $$a\_{1}$$ | $$a\_{0}$$ | $$A\_{1}$$ | $$A$$ |
| 1 | -1 | -1 | 1 | 3 | 3 |
| -1 | -1 | 1 | -1 | -11 | 6 |
| -1 | 1 | -1 | 1 | -5 | -5 |
| 1 | -1 | 1 | 1 | 7 | 7 |
| -1 | 1 | 1 | -1 | -3 | -3 |
| 1 | 1 | -1 | 1 | 11 | -6 |
| 1 | -1 | 1 | -1 | 5 | 5 |
| -1 | 1 | -1 | -1 | -7 | -7 |
| 1 | -1 | -1 | 1 | 3 | 3 |

Это следует из того, что элемент -1 является корнем уравнения (12), рассматриваемого при $z=x^{8}$ или из того, что в рассматриваемой области $2^{4}≡-1\left(17\right)$.

Следовательно, операция умножения на два при использовании знакопеременного двоичного представления числа сводится к циклической перестановке двоичных элементов в записи, вытекающей из (18), с изменением знака переставляемого элемента. Имеем

$2∙A=2∙a\_{2}a\_{1}a\_{0}a\_{3}=a\_{2}a\_{1}a\_{0}\left(-a\_{3}\right)$ (29)

Видно, что это свойство действительно аналогично свойству, которым обладает операция умножения на два в полях, образованных с помощью чисел Мерсенна (2).

Свойство (29), в частности, позволяет реализовать алгоритм умножения двух элементов поля $GF\left(17\right)$представленного в знакопеременной двоичной форме, аналогичный алгоритму, задаваемому формулами (6), (7).

Действительно, результат умножения числа $A$, представленного в знакопеременной форме (26), на число $B$представленное в той же форме

$B=2^{3}∙b\_{3}+2^{2}∙b\_{1}+2^{1}∙b\_{1}+2^{0}∙b\_{0}$ (30)

это сумма следующих терминов

$A∙b\_{0}=\left(2^{3}∙a\_{3}+2^{2}∙a\_{2}+2^{1}∙a\_{1}+2^{0}∙a\_{0}\right)b\_{0}$ (31)

$2^{1}∙A∙b\_{1}=\left(2^{3}∙a\_{2}+2^{2}∙a\_{1}+2^{1}∙a\_{0}-2^{0}∙a\_{3}\right)b\_{1}$ (32)

$2^{2}∙A∙b\_{2}=\left(2^{3}∙a\_{1}+2^{2}∙a\_{0}-2^{1}∙a\_{3}-2^{0}∙a\_{2}\right)b\_{2}$ (33)

$2^{3}∙A∙b\_{3}=\left(2^{3}∙a\_{0}-2^{2}∙a\_{3}-2^{1}∙a\_{2}-2^{0}∙a\_{1}\right)b\_{3}$ (34)

Группируя элементы по степеням двойки, мы получим следующие выражения для коэффициентов $c\_{i}$, которые возникают при умножении двух элементов поля

$c\_{1}=a\_{3}b\_{0}+a\_{2}b\_{1}+a\_{1}b\_{2}+a\_{0}b\_{3}$ (35)

$c\_{2}=a\_{2}b\_{0}+a\_{1}b\_{1}+a\_{0}b\_{2}-a\_{3}b\_{3} $ (36)

$c\_{3}=a\_{1}b\_{0}+a\_{0}b\_{1}-a\_{3}b\_{2}-a\_{2}b\_{3}$ (37)

$c\_{4}=a\_{0}b\_{0}-a\_{3}b\_{1}-a\_{2}b\_{2}-a\_{1}b\_{3} $ (38)

Видно, что в формулах (35) - (38) коэффициенты $a\_{i}$ циклически переставляются с изменением знака, что соответствует специфике рассматриваемого поля. Подчеркнем также, что коэффициенты $c\_{i}$ не обязательно равны $\pm 1$т.е. не являются коэффициентами в представлении вида (26). Это веса, с которыми складываются силы двух при вычислении результата произведения.

Алгоритм, основанный на формулах (35) - (38), может быть реализован достаточно просто схематически, более того, он допускает очевидное обобщение на любые поля $GF\left(2^{n}+1\right)$. Однако тот факт, что коэффициенты $c\_{i}$ не являются коэффициентами в двоичном знакопеременном представлении, заставляет нас задуматься о дальнейшем упрощении схемной реализации умножителей в полях рассматриваемого типа.

**2.6 Алгоритмические основы цифрового логарифма в** $GF\left(2^{n}+1\right)$ **поле**

Значительное упрощение схемной реализации умножителя в рассматриваемой области может быть достигнуто за счет использования операции цифрового логарифма, как отмечалось выше.

Применительно к рассматриваемой области $GF\left(17\right)$ рассматриваемой области, операция цифрового логарифма сводится к операции схемной идентификации примитивных элементов. Как видно из таблицы 2, умножение на примитивный элемент приводит к тому, что число становится равным степени двойки, что легко определить схематически при использовании двоичного представления.

Для выявления примитивных элементов можно использовать следующий прием, который вытекает из свойства циклической перестановки с изменением знака переставляемого элемента. В таблице 5 представлены примеры продолжения последовательности знаков $a\_{i}$, появляющихся в знакопеременном двоичном представлении элементов поля $GF\left(17\right)$. Каждая из этих последовательностей может включать один дополнительный элемент -.$a\_{3}$, как показано в таблице 5.

Сформируем функцию $Q\_{j,j-1}$, которая обеспечивает подсчет количества знаковых переменных в последовательностях, аналогичных представленным в таблице 5. Эта функция определяется следующим образом

$q\_{i}=Q\_{j,j-1}=\left\{\begin{matrix}1, a\_{j}a\_{j-1}=-1\\0, a\_{j}a\_{j-1}=+1\end{matrix}\right.$ (39)

Значения $q\_{i}$для непримитивных элементов поля приведены в таблице 6, а для примитивных элементов - в таблице 7.

Таблица 5. Пример продолжения последовательности двоичных символов с поочередным представлением элементов поля $GF\left(17\right)$ поля

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| $$a\_{3}$$ | $$a\_{2}$$ | $$a\_{1}$$ | $$a\_{0}$$ | $$-a\_{3}$$ |
| 1 | -1 | -1 | -1 | -1 |
| -1 | -1 | -1 | -1 | 1 |
| -1 | -1 | -1 | 1 | 1 |

Данные таблицы показывают, что суммы $\sum\_{}^{}q\_{i}$ являются инвариантными для множества примитивных элементов поля $GF\left(17\right)$ (все они, при использовании знакопеременного двоичного представления элементов этого поля, образуются циклической перестановкой со сменой знака), а также для множества элементов, не являющихся примитивными. В одном случае инвариант равен 1, в другом - 3.

Таблица 6. Значения $q\_{i}$ и их суммы для элементов поля $GF\left(17\right)$ поля, которые не являются примитивными

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| $$q\_{3}=Q\_{3,2}$$ | $$q\_{2}=Q\_{2,1}$$ | $$q\_{1}=Q\_{1,0}$$ | $$q\_{0}=Q\_{0,3}$$ | $$\sum\_{}^{}q\_{i}$$ |
| 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |

Таблица 7. Значения $q\_{i}$ и их суммы для примитивных элементов поля $GF\left(17\right)$поля

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| $$q\_{3}=Q\_{3,2}$$ | $$q\_{2}=Q\_{2,1}$$ | $$q\_{1}=Q\_{1,0}$$ | $$q\_{0}=Q\_{0,3}$$ | $$\sum\_{}^{}q\_{i}$$ |
| 1 | 0 | 1 | 1 | 3 |
| 0 | 1 | 1 | 1 | 3 |
| 1 | 1 | 1 | 0 | 3 |
| 1 | 1 | 0 | 1 | 3 |
| 1 | 0 | 1 | 1 | 3 |
| 0 | 1 | 1 | 1 | 3 |
| 1 | 1 | 1 | 0 | 3 |
| 1 | 1 | 0 | 1 | 3 |

Этот подход, основанный на вычислении инвариантов, легко обобщается на другие области $GF\left(2^{n}+1\right)$что непосредственно следует из того, что умножение на два сводится к циклической перестановке со сменой знака. В частности, ненулевые элементы поля $GF\left(257\right)$ распадаются на 16 подмножеств, каждое из которых образовано циклическими перестановками рассматриваемого типа. Это, в свою очередь, следует из того, что в поле $GF\left(257\right) $существует

$2^{8}≡-1\left(257\right)$ (40)

**2.7 Выводы второй статьи**

Таким образом, наряду с прямым алгоритмом умножения чисел в знакопеременном двоичном представлении, задаваемым формулами (35) - (38), мы можем предложить следующий алгоритм выполнения операции умножения для полей $GF\left(2^{n}+1\right)$, в частности, для поля $GF\left(17\right)$.

1. Перемноженные числа переводятся в двоичное знакопеременное представление. Это обеспечивается прямым пересчетом по формуле $A=2A\_{0}+1$где $A\_{0}$исходное число, а затем представление полученного нечетного числа в виде (26).
2. Используя двоичные чередующиеся коэффициенты числа $A=2A\_{0}+1$вычисляются инварианты $\sum\_{}^{}q\_{i}$ вычисляются инварианты, определяющие разбиение поля на подмножества, каждое из которых соответствует умножению на степень двойки.
3. Умножение производится на элемент, соответствующий инварианту $\sum\_{}^{}q\_{i}$. В результате образуется элемент, представляющий собой степень двойки (с учетом знака). Эта степень задаёт цифровой логарифм рассматриваемого элемента поля.
4. Экспоненты складываются по модулю степени двойки, что схематически соответствует обычной операции сложения двоичных чисел с отбрасыванием старшего разряда.
5. Обратный переход от экспоненты к ненулевому элементу поля $GF\left(2^{n}+1\right)$ осуществляется.

В целом, в работе показано, что использование знакопеременного двоичного представления для элементов полей $GF\left(2^{n}+1\right)$ позволяет реализовать все те же преимущества, что и при работе с полями Галуа, соответствующими Мерсенновским приемам.

Наиболее важным типом полей этого типа является поле $GF\left(257\right)$ поскольку оно соответствует количеству уровней оцифрованного сигнала, что часто используется на практике. Например, наиболее распространенные аналого-цифровые преобразователи предполагают использование 256 уровней.

**2.8 Список использованной литературы второй статьи**

1. Lehnigk-Emden T., When N. Complexity evaluation of non-binary Galois field LDPC code decoders // 2010 6th International Symposium on Turbo Codes & Iterative Information Processing, IEEE. – 2010. – P. 53-57.
2. Isla H., Prakash O. New Quantum and LCD Codes over Finite Fields of Even Characteristic // Defence Science Journal. – 2021. – Vol. 71(5).
3. Kuo Y. M., Garcia-Herrero F., Ruano O., Maestro J.A. RISC-V Galois Field ISA Extension for Non-Binary Error-Correction Codes and Classical and Post-Quantum Cryptography // IEEE Transactions on Computers. – 2022.
4. Matsumine T., Ochiai H. A design of non-binary turbo codes over finite fields based on Gaussian approximation and union bounds // 2018 IEEE 87th Vehicular Technology Conference (VTC Spring). – 2021. – P. 1-5.
5. Moldakhan I., Matrassulova D. K., Shaltykova D. B., Suleimenov I. E. Some advantages of non-binary Galois fields for digital signal processing // Indonesian Journal of Electrical Engineering and Computer Science. – 2021. – Vol. 23(2). – P. 871-877.
6. Vitulyova E. S., Matrassulova D. K., Suleimenov I. E. New application of non-binary Galois fields Fourier transform: Digital analog of convolution theorem // Indonesian Journal of Electrical Engineering and Computer Science. – 2021. – Vol. 23(3). – P. 1718-1726.
7. Kalimoldayev M., Tynymbayev S., Gnatyuk S., Ibraimov M., Magzom M. The device for multiplying polynomials modulo an irreducible polynomial // News of the National Academy of Sciences of the Republic of Kazakhstan Series of Geology and Technical Sciences. – 2019. – Vol. 2(434). – P. 199-205.
8. Dey S., Ghosh R. 4-bit crypto S-boxes: Generation with irreducible polynomials over Galois field GF(24) and cryptanalysis // Cryptology ePrint Archive. – 2018.
9. Kuppuswamy P., Al-Khalidi S. Q. Implementation of security through simple symmetric key algorithm based on modulo 37 // International Journal of Computers & Technology. – 2012. – Vol. 3(2). – P. 335-338.
10. Verkhovsky B. S. Enhanced Euclid Algorithm for Modular Multiplicative Inverse and Its Application in Cryptographic Protocols // Int. J. Commun. Netw. Syst. Sci. – 2010. – Vol. 3(12). – P. 901-906.
11. Gabrielyan O.A., Vitulyova Ye., Suleimenov S., I. E. Multi-valued logics as an advanced basis for artificial intelligence // Wisdom. – 2022. – Vol. 1(21). – P. 170-181.
12. Vitulyova Y.S., Bakirov A.S., Baipakbayeva S.T., Suleimenov I.E. Interpretation of the category of complex in terms of dialectical positivism // IOP Conference Series: Materials Science and Engineering. – 2020. – Vol. 946(1). – P. 012004. – DOI: https://doi.org/10.1088/1757-899X/946/1/012004.
13. Gabrielyan O.A., Vitulyova Ye. S., Suleimenov I. E. Multi-valued logics as an advanced basis for artificial intelligence // Wisdom. – 2022. – Vol. 1(21). – P. 170-181.
14. Suleimenov I. E., Matrassulova D. K., Moldakhan I., Vitulyova Y. S., Kabdushev S. B., Bakirov A. S. Distributed memory of neural networks and the problem of the intelligence’s essence // Bulletin of Electrical Engineering and Informatics. – 2022. – Vol. 11(1). – P. 510-520.
15. Lukasiewicz J. On Three-Valued Logic // Jan Lukasiewicz. Selected Works / Ed. by Borkowski L. – Amsterdam: North-Holland, 1970. – P. 87-88.
16. Zamansky A. On recent applications of paraconsistent logic: an exploratory literature review // Journal of Applied Non-Classical Logics. – 2019. – Vol. 29(4). – P. 382-391.
17. Hernández-Tello A., Macías V. B., Coniglio M. E. Paracomplete Logics Dual to the Genuine Paraconsistent Logics: The Three-valued Case // Electronic Notes in Theoretical Computer Science. – 2019. – Vol. 354. – P. 61-74.
18. Nakayama Y., Akama S., Murai T. Deduction System for Decision Logic Based on Many-valued Logics // International Journal on Advances in Intelligent Systems. – 2018. – Vol. 11(1/2). – P. 115-126.
19. Suleimenov I. E., Vitulyova Y. S., Kabdushev S. B., Bakirov A. S. Improving the efficiency of using multi-valued logic tools // Scientific Reports. – 2023. – Vol. 13(1). – P. 1108. – DOI: https://doi.org/10.1038/s41598-023-28272-1.
20. Shahov V.V. Review and comparative analysis of pseudo-random number generator libraries // Problems of computer science. – P. 66-74.
21. Matsumoto M., Nishimura T. Mersenne Twister: A 623-dimensionally equidistributed uniform pseudo-random number generator // ACM Trans. Model. Comput. Simulat. – 1998. – Vol. 8(1). – P. 3-30.
22. Smirnov A. A., Bondar V. V., Rozhenko O. D., Mirzoyan M. V., Darjania A. D. Mersenne Numbers in the Bases of Systems of Residual Classes when Transmitting Data in Serial Communication Channels // Journal of Mathematical Sciences. – 2022. – Vol. 260. – P. 241-248.
23. Kumar R., Jaiswal R. K., Mishra R. A. Perspective and Opportunities of Modulo 2n−1 Multipliers in Residue Number System: A Review // Journal of Circuits, Systems and Computers. – 2020. – Vol. 29(11). – P. 2030008.
24. Elaskary R. M., Mehana A. H., Fahmy Y., El-Ghoneimy M. Performance of digital predistorter in system with analogue to digital converter // IET Communications. – 2022. – DOI: https://doi.org/10.1049/cmu2.12443.
25. Suleimenov I., Bakirov A., Moldakhan I. Formalization of Ternary Logic for Application to Digital Signal Processing // Energy Management of Municipal Transportation Facilities and Transport. – Springer, Cham, 2019. – P. 26-35.
26. Moldovyan D. N. New form of the hidden logarithm problem and its algebraic support // Bul. Acad., Stiin¸te Repub. Mold. Mat. – 2020.
27. Cheng T., Masuda Y., Chen J., Yu J., Hashimoto M. Logarithm-approximate floating-point multiplier is applicable to power-efficient neural network training // Integration, Elsevier. – 2020. – Vol. 74. – P. 19-31.

**3 Особенности поля Галуа GF(257) и его использование для цифровой обработки сигналов**

**3.1 Аннотация третьей статьи**

Предложен алгоритм вычисления цифрового логарифма для поля Галуа $GF(257)$ предложен. Показано, что это поле сопряжено с одним из важнейших существующих стандартов, использующим цифровое представление сигнала по 256 уровням. Показано, что для этого случая целесообразно использовать специфику квазимерсенновских простых чисел, представимых в виде $p=2^{n}+1$в которую входит число 257. Для полей $GF(2^{n}+1)$можно использовать знакопеременное кодирование, при котором ненулевые элементы поля отображаются через двоичные символы, соответствующие числам +1 и -1. В таком кодировании умножение элемента поля на 2 сводится к квазициклической перестановке двоичных символов (переставляемый символ меняет знак). Предложенный подход позволяет существенно упростить конструкцию вычислительных устройств для вычисления цифрового логарифма и умножения чисел по модулю 257. Представлена конкретная схема устройства для вычисления цифрового логарифма в этой области. Также показано, что данная схема может быть оснащена универсальным сумматором по модулю произвольного числа, что позволяет реализовать любые операции в рассматриваемой области. Показано, что предложенный цифровой алгоритм также может быть использован для приведения 256-значных логических операций к алгебраическому виду.

**3.2 Введение** **третьей статьи**

Двоичные и недвоичные поля Галуа находят все большее применение в информационных технологиях. В основном такие алгебраические структуры используются для разработки алгоритмов информационной безопасности [1-3], то есть в той области, которая явно оперирует с буквенно-цифровыми символами, которым можно поставить в соответствие элементы того или иного дискретного множества, изучаемого абстрактной алгеброй. В этой области были получены весьма значительные результаты, использующие, в частности, преобразования Фурье-Галуа [4-6]. Анализ современной литературы в этой области, в частности, таких работ, как [7-9], позволяет утверждать, что теория кодирования (в том числе теория кодов с коррекцией ошибок [10]) уже во многом стала частью прикладной абстрактной алгебры. Для разработки алгоритмов информационной безопасности используются и другие нетривиальные алгебраические структуры, в частности, конечные алгебраические кольца [11-13].

Уместно подчеркнуть, что потребности практики, в частности связанные с вычислением преобразований Фурье-Галуа [14], заставляют обращаться и к использованию нетривиальных кодовых систем [15].

Однако эта тенденция позволяет нам несколько шире взглянуть на прикладное использование инструментов абстрактной алгебры [16]. А именно, в естественнонаучной традиции физические явления, как правило, описываются функциями, принимающими вещественные или комплексные значения. Однако, как подчеркивается в [17,18], это не более чем вопрос соглашения. Функция, принимающая значения в определенном множестве, есть не что иное, как модель реального процесса, например, сигнала [17,18]. С общеметодологической точки зрения, происходит отображение реального физического процесса (например, генерации сигнала) на некоторый математический объект, выбор которого в конечном итоге определяется вопросами удобства и эффективности использования.

В частности, для цифрового сигнала, изменяющегося в конечном диапазоне амплитуд, в качестве модели сигнала может быть использована функция, принимающая значения в полях Галуа [17,18] или алгебраических кольцах [19,20]. Число дискретных уровней, укладывающихся в конечный диапазон амплитуд, также конечное. Следовательно, для описания процессов такого рода может быть использована функция, принимающая значения в любой алгебраической структуре с конечным числом элементов.

Элементам поля Галуа можно присвоить значения многозначной логической переменной. Более того, как показано в [16,21], можно свести любые логические операции к алгебраической форме. Подчеркнем, что традиционно функции, зависящие от логической переменной, представляются в табличной форме [22,23]. Результаты, полученные в [16,21], позволяют преобразовать такие таблицы к явным алгебраическим выражениям.

В этой связи представление функций, описывающих реальные физические процессы, через поля Галуа (элементы которых могут быть отнесены к значениям переменных многозначной логики) создает дополнительные преимущества. Такое перспективное представление позволяет выявить логические связи между параметрами или функциями, описывающими реальные процессы.

Для преобразования в цифровую форму любых параметров, характеризующих любой физический процесс, в основном используются аналого-цифровые преобразователи, соответствующие существующим стандартам. Один из наиболее часто используемых стандартов делит амплитудный диапазон на 256 уровней [24]. Этот стандарт соответствует полю Галуа $GF\left(2^{8}\right)$т.е. каждый из уровней сигнала может быть связан с элементом этого поля.

Следовательно, допустимо рассмотреть проблему сведения вычислений, соответствующих такому количеству дискретных уровней, к вычислениям в поле Галуа. Уместно отметить, что в современной литературе имеются сообщения, посвященные разработке электронных схем, выполняющих операции сложения и умножения по модулю, например, [25,26]. Такие операции соответствуют операциям в простейших полях Галуа $GF\left(p\right)$где $p$ является простым числом. Предлагаемая постановка задачи полностью соответствует этой тенденции.

С точки зрения прикладных целей разные поля Галуа имеют различную специфику [27,28], то есть не всегда оправданно рассматривать вопрос о максимальном обобщении результатов, полученных с использованием конкретного поля Галуа. В частности, поле $GF\left(2^{4}+1\right)$изученное в [29], является сопряженным к полю GF($2^{4}$), что позволяет привести в явный вид любые операции для важного частного случая 16-значной логики. Тот факт, что число 17 является одним из квазимерсеновских простых, представимых в виде $p=2^{n}+1$используется в [29].

Количество уровней, равное 256, как и 16-значная логика, связано с одним из квази-мерсенновских чисел (257), что позволяет существенно упростить любые вычислительные процедуры, связанные с проведением расчетов в поле $GF\left(2^{8}\right)$ за счет перехода к вычислениям в сопряженном поле Галуа $GF\left(257\right)$.

В данной работе представлены конкретные алгоритмы, позволяющие существенно упростить любые вычисления в данной области $GF\left(257\right)$, а также представлены конкретные электронные схемы, доказывающие эффективность предложенных алгоритмов. Важно, что эти алгоритмы, помимо всего прочего, позволяют реализовать вычисления в рассматриваемой области на основе стандартной элементной базы, соответствующей двоичной логике.

В основе этих алгоритмов лежит, в частности, операция цифрового логарифма. Этот вопрос также достаточно активно обсуждается в литературе [30-32], причем есть примеры, когда эта проблема рассматривается применительно к конкретному полю Галуа [33].

В данной работе доказано, что специфика $GF\left(257\right)$ поля позволяет реализовать алгоритм цифрового логарифма, который может быть использован для создания электронных схем, в том числе выполняющих операции в 256-значной логике, т.е. в одном из важнейших технических стандартов.

**3.3 Методы: сравнение чисел Мерсена и квази-Мерсена, используемых для цифровой обработки сигналов**

Метод, используемый в данном отчете, основан на использовании простых чисел, которые можно назвать квази-мерсенскими числами.

Чтобы сравнение было адекватным, кратко рассмотрим основные свойства чисел Мерсенна, которые в настоящее время находят конкретные практические применения [34,35].

Такие числа можно представить в виде

$p=2^{n}-1$ (1)

где $n$ - специально выбранные целые числа, первыми из которых являются 2, 3, 5 и 7.

Применительно к цифровой обработке сигналов представляет интерес следующее свойство чисел Мерсенна, которое удобно рассмотреть на примере поля $GF\left(127\right)$ характеристикой которого является число Мерсенна с $n=7$.

В двоичном виде любой из элементов $GF\left(127\right) $поля может быть представлен как

$A=a\_{6}a\_{5}…a\_{1}a\_{0}$. (2)

Где $a\_{i}=0,1$

Умножение данного числа на 2 по модулю 127 сводится к циклической перестановке символов

$2a\_{6}a\_{5}…a\_{1}a\_{0}=a\_{5}…a\_{1}a\_{0}a\_{6}$ (3)

Формула (3) является следствием следующего соотношения

$1111111≡0000000\left(127\right)$ (4)

Свойство (3), в частности, позволяет существенно упростить алгоритм умножения чисел по модулю 127 друг на друга.

Аналогичный алгоритм для частного случая квази-мерсеновских чисел был предложен в [29].

Такие числа можно представить в виде

$p=2^{n}+1$ (5)

Алгоритм [29] основан на следующих свойствах квази-мерсеновских чисел, которые удобно рассматривать на примере поля $GF\left(17\right)$ в качестве примера. Это поле можно рассматривать как набор элементов

$GF\left(17\right)=\left\{-8,-7,…,0,…7,8\right\}$ (6)

поскольку выбор представляющих элементов произволен, вплоть до операции сравнения по модулю, например, $-8≡9(17)$*.*

Все ненулевые элементы рассматриваемого поля удовлетворяют соотношению

$x^{16}-1=0$ (7)

Из этого соотношения, в частности, следует, что любой элемент рассматриваемого поля может быть представлен в виде

$x=\left(\sqrt[2]{1}\right)^{s\_{4}}\left(\sqrt[4]{1}\right)^{s\_{3}}\left(\sqrt[8]{1}\right)^{s\_{2}}\left(\sqrt[16]{1}\right)^{s\_{1}}$ (8)

где $s\_{i}=0;1$а обозначение $\sqrt[2^{k}]{1}$ обозначает элемент, обладающий следующими свойствами:

$\left[\sqrt[2^{k}]{1}\right]^{2^{k}}=1; \left[\sqrt[2^{k}]{1}\right]^{2^{k-1}}\ne 1 $ (9)

В частности, для поля $GF\left(17\right)$ такие элементы равны

$\left(\sqrt[16]{1}\right)\_{1}=-1$ (10)

$\left(\sqrt[16]{1}\right)\_{2}=4;-4$ (11)

$\left(\sqrt[16]{1}\right)\_{3}=2;8;-8;-2$ (12)

$\left(\sqrt[16]{1}\right)\_{4}=3;5;6;7;-7;-6;-5;-3$ (13)

Возможность использования представления (8) следует из общей теории полей Галуа. Действительно, все мощности любого примитивного элемента $y$ рассматриваемого поля от 0 до 15 различны. В то же время все эти мощности являются корнями уравнения (5), то есть исчерпывают элементы поля.

Рассмотрим степень $y^{m}$и представим число m, где $0\leq m\leq 15$ в двоичной форме

$m=m\_{4}m\_{3}m\_{2}m\_{1}$ (14)

где $m\_{i}$ двоичные символы, т.е.

$m=2^{3}∙m\_{4}+2^{2}∙m\_{3}+2^{1}∙m\_{2}+2^{0}∙m\_{1}$ (15)

Таким образом, степень $y^{m}$ можно представить в виде

$y^{m}=\left(y^{8}\right)^{m\_{4}}\left(y^{4}\right)^{m\_{3}}\left(y^{2}\right)^{m\_{2}}\left(y^{1}\right)^{m\_{1}}$ (16)

Это выражение можно свести к виду (8), используя соотношения (10) - (13).

Начиная с выражения вида (8), можно предложить чередующееся кодирование [29], которое также удобно рассмотреть на примере $GF\left(17\right)$ поля.

Рассмотрим выражение для ненулевых элементов $GF\left(17\right)$ поля

$A=2^{3}∙a\_{3}+2^{2}∙a\_{2}+2^{1}∙a\_{1}+2^{0}∙a\_{0}$ (17)

Где $a\_{i}=\pm 1$

Результат вычислений по формуле (17) обязательно даст нечетное число. Существует $2^{4}$ комбинаций вида (17), максимальное число которых равно $A\_{max}=15$ и, соответственно, $A\_{min}=-15$.

Видно, что число комбинаций вида (17) в рассматриваемом случае совпадает с числом ненулевых элементов поля $GF\left(17\right)$. Таким образом, после редукции по модулю числа, представленные в форме (17), исчерпывают множество ненулевых элементов поля $GF\left(17\right)$. Следовательно, это представление может быть использовано наряду с любым другим, особенно если учесть, что "представители" классов остатков кольца целых чисел по модулю простого числа могут быть выбраны произвольным образом.

Представление вида (17), в котором $a\_{i}=\pm 1$ обладает свойством, аналогичным тому, которым обладают простые числа Мерсенна (3). А именно, умножение числа, записанного в представлении (15), на 2 можно представить в виде следующей операции

$2∙A=2^{3}∙a\_{2}+2^{2}∙a\_{1}+2^{1}∙a\_{0}-2^{0}∙a\_{3}$ (18)

Это следует из того, что в рассматриваемой области $2^{4}≡-1\left(17\right)$.

Следовательно, операция умножения на 2 в знакопеременном двоичном представлении числа сводится к циклической перестановке двоичных элементов с изменением знака последнего элемента. Имеем

$2∙A=2∙a\_{2}a\_{1}a\_{0}a\_{3}=a\_{2}a\_{1}a\_{0}\left(-a\_{3}\right)$ (19)

Видно, что это свойство действительно похоже на свойство полей, образованных с помощью чисел Мерсенна (3).

Это свойство дает возможность предложить простой алгоритм операции цифрового логарифма и для очень важного частного случая $GF\left(257\right)$ тоже.

**3.4 Результаты и обсуждение** **третьей статьи**

Вычисление элементов $s\_{j}$ в представлении (6) и подобных ему соответствует операции цифрового логарифма.

С помощью алгоритмов (и/или цифровых устройств), выполняющих такую операцию, операция умножения, очевидно, может быть сведена к операции сложения.

В этом разделе доказано, что множество ненулевых элементов поля GF(257) можно разбить на подмножества, и это разбиение позволяет существенно упростить операцию цифрового логарифма. Забегая несколько вперед, отметим, что предложенные алгоритмы также позволяют существенно упростить электронные схемы, выполняющие эту операцию.

Предлагаемый алгоритм основан на величинах, которые можно назвать вычислительными инвариантами. Обоснование их использования приведено в этом разделе.

Начнем с тождества

$x^{256}-1=\left(x^{16}\right)^{16}-1$, (20)

Правая часть соотношения (20) подчеркивает следующий факт. Для представления произвольного ненулевого элемента поля GF(257) с помощью соотношения (17) требуется 8 бит. С учетом изменения знака последнего элемента при умножении на 2 (16), существует 16 элементов, отличающихся друг от друга коэффициентом $2^{k}$, k=0,1,...,15 в этом поле.

Поэтому произвольный ненулевой элемент данного поля может быть представлен в виде

$x=\left(\sqrt[16]{2}\right)^{σ}2^{k}$ (21)

где $σ=0,1,…, 15$, $k=0,1,…,15$.

Подчеркнем, что корень $\sqrt[16]{2}$ должен быть выбран равным примитивному элементу, т.е. различные степени корня должны давать все элементы поля $GF\left(257\right)$.

Аналогично, все ненулевые элементы поля $GF\left(17\right)$ могут быть выражены формулой, аналогичной (24), полученной в [29]:

$x=\left(\sqrt{2}\right)^{σ}2^{k}$ (22)

Адекватность представлений (21) также можно доказать следующим образом. Рассмотрим произвольную степень примитивного элемента $\sqrt[16]{2}$

$z=\left(\sqrt[16]{2}\right)^{w}$ (23)

где $w=0,1,…,256$.

Представим число *w* в стандартной двоичной кодировке

$w=2^{7}w\_{7}+2^{6}w\_{6}+…+2^{0}w\_{0}$ (24)

Подставим выражение (24) в формулу (23). Имеем

$z=\left(2^{4}\right)^{w\_{7}}\left(2^{3}\right)^{w\_{6}}…\left(\sqrt[8]{2}\right)^{w\_{1}}\left(\sqrt[16]{2}\right)^{w\_{0}}$ (25)

Отметим, что формула (25) одинаково справедлива для представления элементов поля $GF\left(257\right)$ как в виде -128, -127 ,...,0,..., 127, 128, так и в виде 0,..., 255, 256. Эти представления отличаются только тем, что в качестве "представителей" соответствующих классов остатков используются $\sqrt[i]{2}$ (табл. 1 и 2).

Существует ровно 16 примитивных элементов $\sqrt[16]{2}$ в поле $GF\left(257\right)$. Они перечислены в таблице 1. В таблице 2 представлены только 9 элементов, так как в случае знакопеременного представления элементы $\sqrt[16]{2}$ делятся на две группы, различающиеся по знаку. Это подчеркивается 9-й строкой в таблице 2.

Таблица 1. Элементы $\sqrt[i]{2}$ поля $GF\left(257\right)$ в терминах положительных чисел

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| $$\sqrt[16]{2}$$ | $$\sqrt[8]{2}$$ | $$\sqrt[4]{2}$$ | $$\sqrt[2]{2}$$ | $$\left(\sqrt[16]{2}\right)^{16}$$ |
| 27 | 215 | 222 | 197 | 2 |
| 41 | 139 | 46 | 60 | 2 |
| 54 | 89 | 211 | 60 | 2 |
| 71 | 158 | 35 | 197 | 2 |
| 82 | 42 | 222 | 197 | 2 |
| 93 | 168 | 211 | 60 | 2 |
| 108 | 99 | 35 | 197 | 2 |
| 115 | 118 | 46 | 60 | 2 |
| 142 | 118 | 46 | 60 | 2 |
| 149 | 99 | 35 | 197 | 2 |
| 164 | 168 | 211 | 60 | 2 |
| 175 | 42 | 222 | 197 | 2 |
| 186 | 158 | 35 | 197 | 2 |
| 203 | 89 | 211 | 60 | 2 |
| 216 | 139 | 46 | 60 | 2 |
| 230 | 215 | 222 | 197 | 2 |

Таблица 2. Элементы $\sqrt[i]{2}$ из $GF\left(257\right)$ поля в представлении с использованием положительных и отрицательных чисел

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| $$\sqrt[16]{2}$$ | $$\sqrt[8]{2}$$ | $$\sqrt[4]{2}$$ | $$\sqrt[2]{2}$$ | $$\left(\sqrt[16]{2}\right)^{16}$$ |
| 27 | -42 | -35 | -60 | 2 |
| 41 | -118 | 46 | 60 | 2 |
| 54 | 89 | -46 | 60 | 2 |
| 71 | -99 | 35 | -60 | 2 |
| 82 | 42 | -35 | -60 | 2 |
| 93 | -89 | -46 | 60 | 2 |
| 108 | 99 | 35 | -60 | 2 |
| 115 | 118 | 46 | 60 | 2 |
| -115 | 118 | 46 | 60 | 2 |

Таким образом, формула (25) показывает, что до перестановки указанного типа знакопеременное двоичное представление позволяет свести все элементы поля $GF\left(257\right)$ к шестнадцати элементам вида

$y=\left(\sqrt[2]{2}\right)^{w\_{3}}\left(\sqrt[4]{2}\right)^{w\_{2}}\left(\sqrt[8]{2}\right)^{w\_{1}}\left(\sqrt[16]{2}\right)^{w\_{0}}$ (26)

Остальные элементы $GF\left(257\right)$ могут быть получены из этих шестнадцати элементов путем циклической перестановки с изменением знака символов в чередующемся двоичном представлении.

Кроме того, представление элементов поля $GF\left(257\right) $в чередующемся двоичном кодировании позволяет сформировать функцию $Q\_{j,j-1}$.

$q\_{i}=Q\_{j,j-1}=\left\{\begin{matrix}1, a\_{j}a\_{j-1}=-1\\0, a\_{j}a\_{j-1}=+1\end{matrix}\right.$ (27)

Эта функция обеспечивает подсчет количества смен знака в кодовых последовательностях, соответствующих знакопеременному представлению элемента поля. В рассматриваемом случае эта функция содержит 8 тактов.

В противном случае можно считать, что эта функция "принимает значения" в вершинах восьмиугольника, как показано на рис. 1. Умножение на степень двойки в этом геометрическом представлении соответствует повороту восьмиугольника вокруг оси симметрии на угол, кратный $\frac{π}{4}$.

Следовательно, значения (27) на самом деле соответствуют вполне определенным геометрическим конструкциям (рис. 1). В связи с тем, что используются циклические перестановки с изменением знака, количество ненулевых значений $q\_{i}$ функции (27) может быть только нечетным.



Рисунок 1 – Геометрическая интерпретация элементов, задаваемых формулой (27); черные кружки - вершины, соответствующие значениям, равным 1 в формуле (27), черные кружки с дополнительным черным кружком соответствуют случаю, когда вершины, соответствующие таким значениям, разделены одной вершиной

Эта геометрическая интерпретация проиллюстрирована в таблице 3. Она показывает, что для каждого конкретного числа, представленного в чередующемся кодировании, существуют определенные инварианты, которые соответствуют ситуациям, представленным на рис. 1.

Таблица 3. Инварианты последовательностей, соответствующих чередующемуся кодированию (примеры).

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| n | q7 | q6 | q5 | q4 | q3 | q2 | q1 | q0 | Σqi |
| -255 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| -253 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| -251 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 3 |
| -249 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| -247 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 3 |
| -245 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 3 |
| -243 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 3 |
| -241 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| -239 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 3 |
| -237 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 3 |
| -235 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 5 |
| -233 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 3 |
| -231 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 3 |
| -229 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 3 |
| -227 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 3 |
| -225 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| -223 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 3 |
| -221 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 3 |
| -219 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 5 |
| -217 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 3 |
| -215 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 5 |
| -213 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 5 |
| -211 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 5 |
| -209 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 3 |
| -207 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 3 |

В этой таблице приведены примеры $q\_{i}$ значений для различных элементов поля $GF\left(257\right)$ в знакопеременном двоичном кодировании, а также инварианты, соответствующие количеству смен знака в последовательности (последний столбец таблицы 3).

Видно, что примеры, представленные в этой таблице, четко соотносятся с геометрической конструкцией рис. 1. А именно, количество смен знака в рассматриваемых последовательностях нечетное, а значит, может быть равно 1, 3, 5 и 7. Последние два случая (инварианты 5 и 7) сводятся к первым двум из тех, на которые указывает инверсия 0↔1.

Следовательно, только случаи $\sum\_{}^{}q\_{i}=1$ и $\sum\_{}^{}q\_{i}=3$ можно не рассматривать. Они соответствуют возможным размещениям одного и трех элементов на вершинах восьмиугольника, как показано на рис. 1, причем такое размещение задается вплоть до поворота на угол $\frac{π}{4}$т.е. размещения, отличающиеся при повороте на такой угол, считаются одинаковыми.

Предложенные инварианты позволяют существенно упростить операцию цифрового логарифма и предложить конкретную электронную схему, реализующую эту операцию на основе стандартных логических элементов. Она рассматривается в следующем разделе.

Как следует из материалов предыдущего раздела, задача вычисления цифрового логарифма в поле $GF\left(257\right)$ делится на две.

Первый из них эквивалентен определению угла поворота восьмиугольника, изображенного на рис. 1, что фактически соответствует определению силы двух (при умножении по модулю 257) в представлении (25).

Вторая соответствует идентификации одной из конфигураций, представленных на рис. 1. Определив такую конфигурацию, можно автоматически определить один из элементов, представленных формулой (26).

Рассмотрим блок-схему устройства для вычисления цифрового логарифма в соответствии с алгоритмом, который следует из вышесказанного. Рассмотрим инварианты второго порядка, отражающие взаимное расположение единиц в вершинах восьмиугольника (рис. 1). Такими инвариантами являются, в частности, суммы $U\_{S\_{j}}$

$U\_{S\_{j}}=\sum\_{}^{}q\_{i}^{\left(s\_{j}\right)}$ (28)

величины $q\_{i}^{\left(s\_{1,2}\right)}$ заданных формулами

$q\_{i}^{\left(s\_{1}\right)}=q\_{i}q\_{i-1}$ (29)

$q\_{i}^{\left(s\_{2}\right)}=q\_{i}q\_{i-2}$ (30)

Интерес представляет выявление семи случаев, кроме тривиального (рис. 1а). В этом тривиальном случае цифра номера напрямую соответствует ненулевому значению $q\_{j}$ (27).

Видно, что инвариант $U\_{S\_{1}}$ равен 2 только для конфигурации, показанной на рис. 1б. Следовательно, вычисление этого инварианта автоматически дает идентификацию этого случая. В этом случае угол поворота шестиугольника фиксируется функцией (29), которая в данном случае принимает только одно ненулевое значение.

Эта сумма $U\_{S\_{1}}$ также принимает ненулевое значение, равное 1, для случаев рис. 1c - 1f. Это соответствует тому, что в последовательности есть только два узла, расположенных в непосредственной близости друг от друга. Для случаев рис. 1g и 1h этот инвариант равен нулю. Следовательно, вычисление указанного инварианта позволяет идентифицировать случай, соответствующий рис. 1b, и разделить оставшиеся варианты на два подмножества.

Вычисление инварианта $U\_{S\_{2}}$ автоматически выбирает два случая из рассматриваемого множества (рис. 1d и 1e). На этих диаграммах нет вершин, отделенных друг от друга только одной пустой вершиной. Кроме того, вычисление этого инварианта однозначно идентифицирует случай рис. 1г, для которого этот инвариант равен 2. Для наглядности заполненные вершины, отделенные друг от друга двумя поворотами под углом π/4, выделены дополнительными кружками.

Таким образом, для завершения операции цифрового логарифмирования для рассматриваемого поля остается только обеспечить различие между случаями рис. 1c и рис. 1f, а также между случаями рис. 1d и рис. 1e. Это различие определяется сдвигом фаз между функциями (29) и (30), который не зависит от наличия коэффициента, равного степени 2 (что эквивалентно поворотам рассматриваемых восьмиугольников).

В частности, этот фазовый сдвиг может быть определен, например, путем вычисления инвариантов, построенных на функциях

$q\_{i}^{\left(s\_{3}\right)}=q\_{i+2}q\_{i}q\_{i-1}$ (31)

$q\_{i}^{\left(s\_{4}\right)}=q\_{i+3}q\_{i}q\_{i-1}$ (32)

Для использования таких функций важно учитывать направление; в частности, предполагается, что на рис. 1 направление отсчитывается по часовой стрелке.

Видно, что функция (31) может принимать ненулевое значение только для случая рис. 1c, а функция (32) - только для случая рис. 1d. Поэтому, чтобы определить указанные случаи, достаточно посчитать инварианты $\sum\_{}^{}q\_{i}^{\left(s\_{3}\right)}$и $\sum\_{}^{}q\_{i}^{\left(s\_{4}\right)}$ .

В результате можно предложить следующую блок-схему устройства для получения цифрового логарифма (рис. 2).



Рисунок 2 – Блок-схема устройства для цифрового логарифмирования в полевых условиях $GF\left(257\right)$

Устройство, построенное по этой схеме, работает следующим образом.

Исходные двоичные сигналы (их количество равно 16) поступают на преобразователь (1), который преобразует их в привычную кодировку. Благодаря этому, в частности, набор сигналов, формируемый на выходе (1), можно считать циклическим.

Со всех 16 выходов преобразователя сигналы поступают на сумматор (2), который подсчитывает количество единиц. На выходе формируется логическая единица $A\_{1}$ сумматора (2), если количество единиц равно 1, то есть реализуется случай, соответствующий рис.1а. В этом случае цифровой логарифм числа в точности равен номеру выхода преобразователя (1), на котором формируется сигнал, отличный от 0.

На выходе формируется логическая единица $A\_{2}$ сумматора (2), если число единиц равно 3, т.е. реализуются случаи, соответствующие рис. 1c-h. Если число единиц равно 5, то на выходе формируется логическая единица $A\_{3}$, а если 7, то на выходе $A\_{4}$.

Сигналы с выходов $A\_{3}$ и $A\_{4}$ поступают на элемент ИЛИ (4). Этот сигнал используется для управления инверторами (5i), которые выполняют операцию инверсии 0↔1 при условии, что число единиц равно 5 или 7. Инверторы (5i) выполняют операцию сложения по модулю 2, т.е. являются элементами ИСКЛЮЧАЮЩЕГО ИЛИ. На вторые входы инверторов (5i) подаются сигналы с выходов преобразователя (1).

В результате на выходах инверторов (5i) формируется набор сигналов, содержащих либо один ненулевой сигнал, либо три таких сигнала.

Этот набор сигналов поступает на первый этап первого блока цифровых логарифмов, состоящий из элементов (6i) и (7i), а также логического элемента ИЛИ (8).

Элементы (6i) выполняют операцию логического ИЛИ. На один из входов каждого из этих элементов подается сигнал с выхода элемента (8), а на второй - сигнал с выхода инвертора (5*i*-1). Следовательно, сигнал на выходе каждого из элементов (6i) будет равен 1, если на выходе сформирована логическая единица $A\_{1}$ или выходе $A\_{4}$ сумматора (2), и равен значению $\left(i-1\right)$st сигнала в противоположном случае.

Сигнал с выхода каждого из элементов (6*i*) подается на один из входов элемента (7*i*), который выполняет операцию логического И. На второй вход каждого из этих элементов сигнал подается непосредственно с выхода инвертора (5*i*).

В результате рассматриваемый блок вычисляет соответствующий инвариант, если число единиц в наборе сигналов на выходе преобразователя (1) было равно 3 или 5, и оставляет сигналы, генерируемые набором преобразователей (5i), без изменений, если указанное число было равно 1 или 7.

Как следует из рис. 1, этот каскад уже идентифицирует последовательности, соответствующие рис. 1а, а также рис. 1b-f в том смысле, что для этих случаев в наборе сигналов, формируемых на выходах элементов (7i), образуется только одна логическая единица, однозначно идентифицирующая цифровой логарифм соответствующего элемента поля Галуа.

Для выявления оставшихся случаев используется сумматор (9), который подсчитывает количество единиц на выходах элементов (7i). Если это число равно 1, то операцию цифрового логарифма следует считать завершенной. Если это число равно 0, то возникают ситуации, соответствующие рис. 1г или рис. 1ч. В этом случае результат цифрового логарифма формируется вторым блоком цифрового логарифма, который рассматривается ниже. Идентификационные сигналы формируются на выходах $B\_{1}$, $B\_{2}$и $B\_{3}$ сумматора (9). На этих выходах формируется логическая единица, если сумма логических единиц на выходах элементов (7i) равна 0, 1 и 2 соответственно.

Для учета ситуации, соответствующей рис. 1б, используется второй этап первого блока цифрового логарифма, который выполнен аналогично первому. Он состоит из элементов (10i) и (11i). Элементы (10i) выполняют операцию логического ИЛИ, а элементы (11i) - операцию И. На первые входы каждого из элементов (10i) подаются сигналы с выходов элементов (7i-1), а на их вторые входы - сигнал с выхода *B*2 , т.е. в случае, когда на этом выходе стоит логическая единица, то элементы (10i ) не влияют на работу системы.

Если на указанном входе формируется логический ноль, то второй каскад де-факто циклически выполняет операцию

$q\_{i}^{\left(5\right)}=q\_{i+1}q\_{i}q\_{i-1}$ (33)

Эта операция в случае, соответствующем рис. 1b, приводит к появлению логической единицы только на одном из выходов элементов (11i), выходы которых соединены со входами устройства принятия решений (12).

Таким образом, рассматриваемая часть схемы обеспечивает однозначную идентификацию цифрового логарифма для случаев, соответствующих рис.1a-f.

Случаи рис.1 g,h соответствуют логической единице, формируемой на выходе *B*1 сумматора 9. Этот сигнал блокирует работу первого блока цифрового дифференцирования и передает его на второй, устроенный аналогичным образом, с той лишь разницей, что идентификация осуществляется с помощью функций (29) и (30).

Таким образом, мы показали, что использование знакопеременного двоичного представления для элементов $GF\left(257\right)$ поля позволяет реализовать все те же преимущества, что и при использовании полей Галуа, соответствующих простым числам Мерсенна.

Наиболее важным полем этого типа является поле $GF\left(257\right)$поскольку оно соответствует количеству уровней оцифрованного сигнала, что часто используется на практике. Например, наиболее часто используемые аналого-цифровые преобразователи предполагают использование ровно 256 уровней.

Предложенный алгоритм позволяет свести операцию умножения к операции сложения.

Следовательно, чтобы обеспечить любые вычисления в рассматриваемой области, необходимо также обеспечить выполнение операции сложения по модулю 257.

В следующем разделе показано, что можно реализовать универсальный сумматор для заданного модуля. В частности, он может быть использован для вычислений в рассматриваемой области.

Любые операции, выполняемые в полях Галуа, могут быть сведены к операциям сложения и умножения. Как следует из материалов предыдущих разделов, для выполнения операций умножения с использованием цифровых логарифмов очень важна специфика конкретного поля Галуа. В отличие от этого, операция сложения, как показано ниже, может быть унифицирована.

Еще раз подчеркнем, что схемы сумматоров в полях Галуа известны, в том числе и в патентной литературе, например, [36,37]. Однако известные схемы очень громоздки. Переход к схеме, подобной флип-флопу (в том смысле, что она имеет два устойчивых состояния), позволяет не только упростить схему, но и обеспечить ее унификацию для модулярного сложения.

Ниже мы рассмотрим схему сумматора, основанную на использовании обратной связи между старшим и младшим разрядами в схеме, построенной на использовании обычных однобитных сумматоров.

Предлагаемый модулярный сумматор с настраиваемым значением модуля основан на следующих соотношениях.

Любое целое число p, включая простое, может быть представлено в виде

$p=2^{n}-k$ (34)

Использование сумматоров, предназначенных для выполнения вычислений в терминах полей Галуа, предполагает, что два числа $a\_{1,2}$ суммируются, удовлетворяя строгому неравенству

$a\_{1,2}<2^{n}-k$ (35)

Поэтому следующая сумма будет удовлетворять строгому неравенству

$a\_{1}+a\_{2}+k<2^{n+1}-k$ (36)

Это неравенство, в частности, означает, что если для отображения чисел требуется n двоичных цифр $a\_{1,2}$то $n+1$ двоичных разрядов, конечно, будет достаточно для отображения суммы $a\_{1}+a\_{2}+k$. На этом обстоятельстве основана схема предлагаемого сумматора (рис. 3).

Схема построена и работает следующим образом. Логические сигналы, соответствующие суммируемым числам в двоичном представлении, подаются на входы одноразрядных двоичных сумматоров (5j), $j=1,2,…,n$: входы побитового переноса этих сумматоров подключены к выходам побитового переноса обычным способом. В результате на выходах сумматоров (5j) формируется набор логических сигналов, соответствующих n младшим битам $n+1$ - битового двоичного числа вида

$C=c\_{n+1}c\_{n}c\_{n-1}…c\_{0}$ (37)

Старший бит $C\_{n+1}$выделенный жирным шрифтом в записи (37), соответствует логическому сигналу, сформированному на выходе переноса битов сумматора (5n).



Рисунок 3 – Схема двоичного сумматора с регулируемым значением модуля на основе обратной связи между битами старшего и младшего порядка

Чтобы преобразовать число (37) в число mod *p* в двоичном представлении, можно поступить следующим образом: прибавить к заданному числу число $k$к данному числу и отбросить старший бит, если он приобретает ненулевое значение. Если после прибавления числа $k$, старший разряд равен нулю, то следует вернуться к исходному значению, то есть к числу

$C=c\_{n}c\_{n-1}…c\_{0}$ (38)

Эта процедура полностью соответствует приведению результата суммирования (37) к числу по модулю $2^{n}-k $ в двоичном представлении. Действительно, если после сложения числа $k$ старший разряд имеет значение 0, то это означает, что число $C$ удовлетворяет неравенству

$C<2^{n+1}-k$ (39)

В этом случае сокращение по модулю $p$ не требуется.

Если значение старшего бита отлично от нуля, то есть равно единице, то отбрасывание старшего разряда и сложение числа $k$ означает, что число $2^{n}-k$ было вычтено из числа (37), что соответствует уменьшению по модулю $p$.

В схемотехническом плане эта процедура осуществляется за счет обратной связи между старшим и младшим разрядами (рис. 3).

А именно, на входы одноразрядных сумматоров (1j) подается набор сигналов, соответствующих младшим двоичным разрядам числа (37), $j=1,2,…,n$. Набор логических сигналов, соответствующих числу$k$ в двоичном представлении или нулю, подается на вторые входы этих сумматоров.

Чтобы переключаться между этими двумя состояниями (добавить $k$ или добавить 0), используются элементы (2j, $j=1,2,…,n$, выполняют операцию логического И. На первые входы элементов (2j) подаются постоянные сигналы с шины данных (3), соответствующие числу k в двоичном представлении.

На вторые входы элементов (2j подается сигнал с выхода элемента (4), который выполняет операцию логического ИЛИ. Следовательно, число k прибавляется к числу, формируемому элементом (4), если на выходе элемента (4) образуется логический ноль, и не прибавляется в обратном случае.

На входы элемента (4) подаются сигналы с управляющего тактового генератора прямоугольных импульсов (7), амплитуда которых соответствует логической единице, а также с выхода одноразрядного сумматора (6), на входы которого, в свою очередь, подаются сигналы с выходов переноса битов сумматоров (1n) и (5n).

Таким образом, если значение старшего бита отлично от нуля (что соответствует логической единице на выходе сумматора (6)), то к числу, образованному по (4), будет добавлено число k.

Однако есть важная особенность использования обратной связи по схеме, приведенной на рис. 3. А именно, рассмотрим число, удовлетворяющее неравенству

$p=2^{n+1}-k\leq C\leq 2^{n}-1$ (40)

Для таких чисел значение старшего разряда в формуле (4) равно нулю, но, тем не менее, их необходимо приводить по модулю, так как они превышают значение $p-1$. Примером может служить случай $C=2^{n+1}-k$, который соответствует ситуации $C≡0(p)$. В этом случае только после сложения k на выходе образуется число 100...0 с ненулевой старшей цифрой.

Чтобы преодолеть эту трудность, схема включает логический элемент (4) и генератор прямоугольных импульсов (7).

В тот момент, когда импульс, снимаемый с генератора (7), активен, на выходе схемы ИЛИ (4) будет сформирована логическая единица, независимо от того, в каком состоянии находился выход сумматора (6). В этом случае *число* $k$ будет добавлено к числу (4).

Если в результате суммирования образуется старший бит, то на выходе сумматора (6) образуется логическая единица, которая останется после прекращения действия импульса, снимаемого с генератора (7). Таким образом, к числу, удовлетворяющему неравенству (7), будет прибавлено число k. Старший разряд отбрасывается автоматически, так как сигналы снимаются только с сумматоров, соответствующих младшим разрядам.

В частности, если в результате суммирования исходных чисел образовалось число $C=2^{n+1}-k$образовалось, то после сложения $k$ обеспечиваемого генератором (7), на выходах сумматоров (1n) образуются логические нули, а на выходе сумматора (6) образуется логическая единица, которая будет поддерживать стационарное сложение числа $k$. Видно, что результирующее состояние выходов (1n) в этом случае действительно соответствует случаю $C≡0(p)$т.е. действительно, сокращение по модулю $p$ выполняется.

Принципиальное отличие этой схемы от схем, представленных в современной патентной литературе, заключается в том, что она фактически является аналогом флип-флопа. Обратная связь между старшим и младшим разрядами, образованная элементами (6), (4) и (2j ), приводит к тому, что для каждого $C$ удовлетворяющего неравенству (40), существует два устойчивых состояния, одно из которых соответствует сравнению по модулю p. Переход из одного состояния в другое обеспечивается дополнительным воздействием - импульсом, формируемым генератором (7), что полностью соответствует записи информации в однобитовую ячейку памяти на основе триггера.

Таким образом, переход к схемам сумматоров, которые также имеют бистабильные состояния, позволяет существенно их упростить.

Рассмотрим возможности обобщения предложенного подхода, хотя на данном этапе исследования этот вопрос представляет скорее академический интерес. В частности, следующим квази-мерсенновским числом после 257 является число 216 +1=65537.

Для любого поля Галуа $GF\left(2^{n}+1\right)$ мы имеем

$2^{n-1}≡-1\left(2^{n}+1\right)$ (41)

Это следует из того, что

$2^{n}+1≡0\left(2^{n}+1\right)$ (42)

Далее, все ненулевые элементы поля $GF\left(2^{n}+1\right) $являются корнями уравнения

$x^{2^{n}}-1=0$, (43)

что непосредственно следует из теории конечных алгебраических полей: число ненулевых элементов поля на единицу меньше их общего числа, поэтому все элементы поля удовлетворяют уравнению степени $2^{n}$.

В соответствии с использованным выше методом, это уравнение удобно представить в следующем виде

$x^{2^{n}}-1=\left(x^{\frac{2^{n}}{2n}}\right)^{2n}-1=0$ (44)

где $N$ количество битов в двоичном знакопеременном представлении поля $GF\left(2^{n}+1\right)$.

Формула (44) учитывает следующий факт. $n$ Для представления ненулевых элементов рассматриваемого поля в знакопеременном кодировании требуются биты,

$N=log\_{2}\left(2^{n}\right)=n$ (45)

Таким образом, в рассматриваемом представлении существуют $2n$ элементов поля, отличающихся друг от друга умножением на степень 2 в рассматриваемом представлении: для того, чтобы квазициклическая перестановка привела к исходному результату, элемент поля должен быть умножен на $2^{2n}$ по модулю $2^{n}+1$. Следовательно, множество ненулевых элементов рассматриваемого поля разбивается на 2n подмножеств, каждое из которых содержит $\frac{2^{n}}{2n}$ элементов.

Значения этих величин для первых 4 квази-мерсенновских чисел представлены в таблице 4.

Таблица 4. Значения величин для первых 4 квази-мерсенновских чисел

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| $$n$$ | 2 | 4 | 8 | 16 |
| $$2^{n}+1$$ | 5 | 17 | 257 | 65537 |
| $$2n$$ | 4 | 8 | 16 | 32 |
| $$2^{n}/2n$$ | 1 | 2 | 16 | 2048 |

Видно, что для полей Галуа, соответствующих первым трем квази-мерсенновским числам, безусловно, имеет смысл выделять подмножества, элементы которых отличаются умножением на степень 2. Уже для четвертого квази-мерсенновского числа целесообразность использования такого подхода, как минимум, не очевидна. Элементов, подобных показанным на рис. 1, становится не 8, а 1024.

В целом, таблица 4 показывает, что предложенный подход действительно целесообразно применять к конкретным полям Галуа *GF*(17) и *GF*(257), представляющим практический интерес. С учетом результатов, полученных в [29], поле *GF*(5) также может представлять интерес как сопряженное (в смысле цифрового логарифма) с *полем* $GF\left(2^{2}\right)$ для упрощения операций четырехзначной логики.

Покажем, что полученные результаты действительно представляют интерес с точки зрения сочетания методов цифровой обработки сигналов и многозначной логики для случая, когда сигнал сводится к 256 дискретным уровням.

Как показано в [16,21], для сведения операций многозначной логики к алгебраическим целесообразно использовать алгебраический аналог δ-функции

$δ\_{i}\left(x\right)=1-\left(x-x\_{i}\right)^{p^{n}-1}$ (46)

где $x$ текущая переменная, принимающая значения в поле Галуа $GF\left(p^{n}\right)$, $x\_{i} $i-й элемент рассматриваемого поля.

Эта функция обладает следующим свойством

$δ\_{i}\left(x\right)=\left\{\begin{matrix}1, x=x\_{i}\\0, x\ne x\_{i}\end{matrix}\right.$(47)

Это свойство следует из общей теории полей Галуа, согласно которой для произвольного элемента поля Галуа, содержащего $p^{n}$ элементы $x^{p^{n}-1}=1$.

Использование алгебраического аналога *δ-функции* позволяет, в частности, свести любые операции к $p^{n}$-логики (логики, в которой число значений переменной равно $p^{n}$) к явному алгебраическому виду [16,21]. Применительно к логической функции двух переменных и полям Галуа *GF(p)* соответствующее выражение имеет вид

$F\left(x,y\right)=\sum\_{i,j=0}^{i,j=p-1}f\left(x\_{i},y\_{j}\right)δ\_{i}\left(x\right)δ\_{j}\left(y\right)$ (48)

где значения $f(x\_{i},y\_{j})$ образуют таблицу истинности (напомним, что операции многозначной логики в настоящее время задаются с помощью таблиц истинности [22]).

Соотношение (48) наглядно выражает известный факт: если число значений переменных многозначной логики равно целой степени простого числа, то логические операции можно свести к вычислениям в сопряженном поле Галуа.

Если это условие не выполняется, то целесообразно модифицировать алгебраический аналог δ-функции с помощью операции цифрового логарифма [29]. Более того, этот подход целесообразно использовать и для случая поля $GF\left(p^{n}+1\right)$.

В частности, мы можем составить алгебраический аналог δ-функции в следующем виде

$δ\_{n\_{i}}\left(n\right)=1-\left(θ^{n}-θ^{n\_{i}}\right)^{p^{n}}$ (49)

В этой формуле, $n$ и $n\_{i}$ обозначают целые числа, соответствующие элементам поля $GF\left(p^{n}\right)$. Предполагается, что значения самой функции $δ\_{n\_{i}}\left(n\right)$ принадлежат полю $GF\left(p^{n}+1\right)$на котором выполняется отображение; $θ$ - примитивный элемент поля $GF\left(p^{n}+1\right)$т.е. таким элементом, степени которого исчерпывают все ненулевые элементы данного поля.

Эта формула удобна тем, что позволяет получить выражение для любой операции, выполняемой в терминах 256-значной логики, к алгебраическим.

Действительно, используя (49), мы имеем

$Q\left(n,m\right)=\sum\_{i,j=0}^{i,j=p-2}Q\_{i,j}δ\_{n\_{i}}\left(θ^{n}\right)δ\_{m\_{j}}\left(θ^{m}\right)$ (50)

где величины $Q\_{i,j}$ связаны с элементами таблицы истинности $p^{n}$- логики, оцениваемой следующим образом

$Q\_{i,j}=θ^{n\_{i,j}}$ (51)

где $n\_{i,j}$ число, соответствующее элементу таблицы истинности с номерами $i,j$.

Формулу (50) можно рассматривать как функцию от пары элементов поля $GF\left(p^{n}\right)$соответствующих числам $n$ и $m$, и принимающих значение в поле $GF\left(p^{n}+1\right)$. Формула (50) допускает очевидное обобщение на функцию произвольного числа элементов из поля $GF\left(p^{n}\right)$.

При подстановке двух произвольных $n\_{0}$ и $m\_{0}$ в формулу (50), в силу (49), имеем

$Q\left(n\_{0},m\_{0}\right)=Q\_{n\_{0},m\_{0}}$ (52)

Только один член в формуле (50) ненулевой.

Видно, что для того, чтобы вернуться к элементам поля $GF\left(p^{n}\right)$ при использовании формулы (50) требуется именно операция цифрового логарифма, которая предложена в данной работе для важного частного случая $GF\left(2^{8}+1\right)$.

Поэтому в будущем возможно создание систем, напрямую оперирующих 256-значной логикой.

**3.5 Заключение третьей статьи**

Таким образом, полученные результаты еще раз показывают, что для прикладного использования крайне важно учитывать специфику конкретных полей Галуа. В частности, это относится к полю $GF(257)$которое соответствует одному из квазимерсенновских праймов, то есть чисел, которые можно представить в виде $p=2^{n}+1$.

Это поле связано с числом 256, которое соответствует одному из важнейших стандартов, используемых в современных цифровых технологиях.

Для чисел, соответствующих полям $GF(2^{n}+1)$удобно использовать знакопеременное кодирование, при котором умножение на число 2 по модулю $p=2^{n}+1$ соответствует квазициклической перестановке двоичных символов, то есть циклической перестановке с изменением знака переставляемого элемента.

Такое кодирование позволяет реализовать простой алгоритм цифрового логарифма, который позволяет свести операцию умножения к операции сложения и т.д.

В частности, для поля $GF(257)$ операция цифрового логарифма упрощается за счет того, что множество ненулевых элементов этого поля можно разбить на 16 подмножеств, элементы которых отличаются друг от друга квазициклической перестановкой двоичных символов. В результате операция цифрового логарифма для данного поля приводит к идентификации элемента по принадлежности к одному из этих подмножеств.

Важно, что операция цифрового логарифма в рассматриваемой области, приводящая операцию умножения к операции сложения, также может быть реализована с помощью относительно простых электронных схем. Соответствующий пример представлен в данной работе.

Важно также, что в работе предложены относительно простые электронные схемы, обеспечивающие сложение по заданному модулю. Поскольку любые операции в поле Галуа приводят к сложению и умножению (с учетом операции нахождения обратного элемента), предложенные схемы обеспечивают выполнение любых операций в $GF(257)$ поле, что соответствует одному из наиболее часто используемых цифровых стандартов.

Предложенная в данной работе операция алгоритма цифрового логарифма также может быть использована для приведения 256-значных логических операций к явному алгебраическому виду.

**3.6 Список использованной литературы третьей статьи**

1. Lehnigk-Emden T., Wehn N. Complexity evaluation of non-binary Galois field LDPC code decoders // 2010 6th International Symposium on Turbo Codes & Iterative Information Processing. – 2010. – P. 53-57. – DOI: https://doi.org/10.1109/ISTC.2010.5613874.
2. Pruss T., Kalla P., Enescu F. Equivalence verification of large Galois field arithmetic circuits using word-level abstraction via Gröbner bases // Proceedings of the 51st Annual Design Automation Conference. – 2014. – P. 1-6. – DOI: https://doi.org/10.1145/2593069.2593134.
3. Jagadeesh H., Joshi R., Rao M. Group Secret-key generation using algebraic rings in wireless networks // IEEE Transactions on Vehicular Technology. – 2021. – Vol. 70(2). – P. 1538-1553. – DOI: https://doi.org/10.1109/TVT.2021.3054031.
4. Liu P., Pan Z., Lei J. Parameter identification of reed-Solomon codes based on probability statistics and Galois field Fourier transform // IEEE Access. – 2019. – Vol. 7. – P. 33619-33630. – DOI: https://doi.org/10.1109/ACCESS.2019.2904718.
5. Girisankar S. B., Nasseri M., Priscilla J., Lin S., Akella V. Multiplier-free implementation of Galois field Fourier transform on a FPGA // IEEE Transactions on Circuits and Systems II: Express Briefs. – 2019. – Vol. 66(11). – P. 1815-1819. – DOI: https://doi.org/10.1109/TCSII.2019.2894361.
6. Huang Q., Tang L., He S., Xiong Z., Wang Z. Low-complexity encoding of quasi-cyclic codes based on Galois Fourier transform // IEEE Transactions on Communications. – 2014. – Vol. 62(6). – P. 1757-1767. – DOI: https://doi.org/10.1109/TCOMM.2014.2316174.
7. Zhang A., Feng K. A unified approach to construct MDS self-dual codes via Reed-Solomon codes // IEEE Transactions on Information Theory. – 2020. – Vol. 66(6). – P. 3650-3656. – DOI: https://doi.org/10.1109/TIT.2020.2963975.
8. Nardo L. G., et al. A reliable chaos-based cryptography using Galois field // Chaos: An Interdisciplinary Journal of Nonlinear Science. – 2021. – Vol. 31(9). – P. 091101. – DOI: https://doi.org/10.1063/5.0061639.
9. Hazzazi M. M., Attuluri S., Bassfar Z., Joshi K. A novel Cipher-Based data encryption with Galois field theory // Sensors. – 2023. – Vol. 23(6). – P. 3287. – DOI: https://doi.org/10.3390/s23063287.
10. Kuo Y.-M., Garcia-Herrero F., Ruano O., Maestro J. A. RISC-V Galois field ISA extension for non-binary error-correction codes and classical and post-quantum cryptography // IEEE Transactions on Computers. – 2022. – Vol. 72(3). – P. 682-692. – DOI: https://doi.org/10.1109/TC.2022.3174587.
11. Bagheri K., Sadeghi M. R. A new non-associative cryptosystem based on NTOW public key cryptosystem and octonions algebra // ACM Communications in Computer Algebra. – 2015. – Vol. 49(1). – P. 13.
12. Markov V. T., Mikhalev A. V., Nechaev A. A. Nonassociative algebraic structures in cryptography and coding // Journal of Mathematical Sciences. – 2020. – Vol. 245(2). – P. 178-196. – DOI: https://doi.org/10.1007/s10958-020-04685-5.
13. Markov V. T., Mikhalev A. V., Kislitsyn E. S. Non-associative structures in homomorphic encryption // Journal of Mathematical Sciences. – 2022. – Vol. 262(5). – P. 735-739. – DOI: https://doi.org/10.1007/s10958-022-05850-8.
14. Deshmukh T. P., Dewalkar V. P. The design approach for fast computation of Fourier transform over a finite field // 2014 International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE). – 2014. – P. 1-4. – DOI: https://doi.org/10.1109/ICGCCEE.2014.6922465.
15. Chernov V. M. Calculation of Fourier-Galois transforms in reduced binary number systems // Computer Optics. – 2018. – Vol. 42(3). – P. 495-500. – DOI: https://doi.org/10.18287/2412-6179-2018-42-3-495-500.
16. Dorigo M., Theraulaz G., Trianni V. Swarm robotics: Past, present, and future [Point of View] // Proceedings of the IEEE. – 2021. – Vol. 109(7). – P. 1152-1165. – DOI: https://doi.org/10.1109/JPROC.2021.3072740.
17. Chung S.-J., Paranjape A. A., Dames P., Shen S., Kumar V. A survey on aerial swarm robotics // IEEE Transactions on Robotics. – 2018. – Vol. 34(4). – P. 837-855. – DOI: https://doi.org/10.1109/TRO.2018.2857475.
18. Liu W., Gao Z. A distributed flocking control strategy for UAV groups // Computer Communications. – 2020. – Vol. 153. – P. 95-101. – DOI: <https://doi.org/10.1016/j.comcom.2020.01.076>.
19. Wang X., Chen G., Gong H., Jiang J. UAV swarm autonomous control based on internet of things and artificial intelligence algorithms // IFS. – 2021. – Vol. 40(4). – P. 7121-7133. – DOI: https://doi.org/10.3233/JIFS-189541.
20. Zheng Y., Huepe C., Han Z. Experimental capabilities and limitations of a position-based control algorithm for swarm robotics // Adaptive Behavior. – 2022. – Vol. 30(1). – P. 19-35. – DOI: https://doi.org/10.1177/10597123320930418.
21. Quesada W. O., Rodriguez J. I., Murillo J. C., Cardona G. A., Yanguas-Rojas D., Jaimes L. G., Calderón J. M. Leader-follower formation for UAV robot swarm based on fuzzy logic theory // In Artificial Intelligence and Soft Computing. – 2018. – Vol. 10842. – P. 740-751. – DOI: https://doi.org/10.1007/978-3-319-91262-2\_65.
22. Hafez A. T., Kamel M. A. Fault-Tolerant control for cooperative unmanned aerial vehicles formation via fuzzy logic // In 2016 International Conference on Unmanned Aircraft Systems (ICUAS). – 2016. – P. 1261-1266. – DOI: https://doi.org/10.1109/ICUAS.2016.7502660.
23. Suleimenov I. E., Vitulyova Y. S., Kabdushev S. B., Bakirov A. S. Improving the efficiency of using multivalued logic tools // Scientific Reports. – 2023. – Vol. 13(1). – P. 1108. – DOI: https://doi.org/10.1038/s41598-023-28272-1.
24. Moldakhan I., Matrassulova D., Shaltykova D., Suleimenov I. Some advantages of non-binary Galois fields for digital signal processing // Indonesian Journal of Electrical Engineering and Computer Science. – 2021. – Vol. 23. – P. 871-878. – DOI: https://doi.org/10.11591/ijeecs.v23.i2.pp871-878.
25. Vitulyova E. S., Matrassulova D. K., Suleimenov I. E. New application of non-binary Galois fields Fourier transform: Digital analog of convolution theorem // Indonesian Journal of Electrical Engineering and Computer Science. – 2021. – Vol. 23(3). – P. 1718-1726. – DOI: https://doi.org/10.11591/ijeecs.v23.i3.pp1718-1726.
26. Matrassulova D. K., Vitulyova Y. S., Konshin S. V., Suleimenov I. E. Algebraic fields and rings as a digital signal processing tool // Indonesian Journal of Electrical Engineering and Computer Science. – 2022. – Vol. 29(1). – P. 206-216. – DOI: https://doi.org/10.11591/ijeecs.v29.i1.pp206-216.
27. Suleimenov I. E., Bakirov A. S., Vitulyova Y. S. Prospects for the use of algebraic rings to describe the operation of convolutional neural networks // In 2022 The 6th International Conference on Advances in Artificial Intelligence. – 2022. – P. 1-7. – DOI: https://doi.org/10.1145/3571560.3571561.
28. Suleimenov I. E., Vitulyova Y. S., Kabdushev S. B., Bakirov A. S. Improving the efficiency of using multivalued logic tools: Application of algebraic rings // Scientific Reports. – 2023. – Vol. 13(1). – P. 22021. – DOI: https://doi.org/10.1038/s41598-023-49593-1.
29. Marcos J. On a Problem of da Costa // In Essays on the Foundations of Mathematics and Logic. – Vol. 2. – Ed. by Sica G. – Monza: Polimetrica, 2005. – P. 53-69.
30. Ciuciura J. A note on Fernández–Coniglio’s hierarchy of paraconsistent systems // Axioms. – 2020. – Vol. 9(2). – P. 35. – DOI: https://doi.org/10.3390/axioms9020035.
31. Díaz De Aguilar J., et al. Characterization of an Analog-to-digital converter frequency response by a Josephson arbitrary waveform synthesizer // Measurement Science and Technology. – 2019. – Vol. 30(3). – P. 035006. – DOI: https://doi.org/10.1088/1361-6501/aafb27.
32. Kalimoldayev M., Tynymbayev S., Gnatyuk S., Ibraimov M., Magzom M. The device for multiplying polynomials modulo an irreducible polynomial // News of the National Academy of Sciences of the Republic of Kazakhstan Series of Geology and Technical Sciences. – 2019. – Vol. 2(434). – P. 199-205. – DOI: https://doi.org/10.32014/2019.2518-170X.55.
33. Dey S., Ghosh R. A Review of Cryptographic Properties of S-Boxes with Generation and Analysis of Crypto Secure S-Boxes // PeerJ Preprints. – 2018. – DOI: https://doi.org/10.7287/peerj.preprints.26452v1.
34. Vitulyova Y. S., Bakirov A. S., Suleimenov I. E. Galois fields for digital image and signal processing: Evidence for the importance of field specificity // In 2022 5th International Conference on Pattern Recognition and Artificial Intelligence (PRAI). – 2022. – P. 637-642. – DOI: <https://doi.org/10.1109/PRA>
35. Matrassulova D. K., Kabdushev S. B., Bakirov A. S., Suleimenov I. E. Algorithm for analyzing rotating images based on the Fourier–Galois transform // In 2023 15th International Conference on Computer Research and Development (ICCRD). – 2023. – P. 204-209. – DOI: https://doi.org/10.1109/ICCRD56364.2023.10080084.
36. Suleimenov I. E., Vitulyova Y. S., Matrassulova D. K. Features of digital signal processing algorithms using Galois fields GF(2^n+1) // PLoS ONE. – 2023. – Vol. 18(10). – P. e0293294. – DOI: https://doi.org/10.1371/journal.pone.0293294.
37. Joux A., Odlyzko A., Pierrot C. The past, evolving present, and future of the discrete logarithm // In Open Problems in Mathematics and Computational Science. – 2014. – P. 5-36. – DOI: https://doi.org/10.1007/978-3-319-10683-0\_2.
38. Vishnoi S., Shrivastava V. A new digital signature algorithm based on factorization and discrete logarithm problem // International Journal of Computer Trends and Technology. – 2012. – Vol. 3(4). – P. 653-657.
39. Fried J., Gaudry P., Heninger N., Thomé E. A kilobit hidden SNFS discrete logarithm computation // In Annual International Conference on the Theory and Applications of Cryptographic Techniques. – 2017. – P. 202-231.
40. Barbulescu R., Bouvier C., Detrey J., Gaudry P., Jeljeli H., Thomé E., et al. Discrete logarithm in GF(2^809) with FFS // In Public-Key Cryptography–PKC 2014: 17th International Conference on Practice and Theory in Public-Key Cryptography. – 2014. – Vol. 17. – P. 221-238. – DOI: https://doi.org/10.1007/978-3-642-54631-0\_13.
41. Matsumoto M., Nishimura T. Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator // ACM Transactions on Modeling and Computer Simulation. – 1998. – Vol. 8(1). – P. 3-30.
42. Smirnov A. A., Bondar V. V., Rozhenko O. D., Mirzoyan M. V., Darjania A. D. Mersenne numbers in the bases of systems of residual classes when transmitting data in serial communication channels // Journal of Mathematical Sciences. – 2022. – Vol. 260(2). – P. 241-248. – DOI: https://doi.org/10.1007/s10958-022-05688-0.
43. Suleimenov I. E., Vitulyova Y. S., Shaltykova D. B., Matrassulova D. K., Bakirov A. S. Pattern recognition methods as a base of development of new instruments for investigations in physical chemistry // In 2022 The 3rd European Symposium on Software Engineering. – 2022. – P. 127-132. – DOI: https://doi.org/10.1145/3571697.3573941.
44. Patent 36236. Adder by module 2^2 – 1 / Mun G. A., Baipakbaeva S. T., Kadyrzhan K. N., Kabdushev Sh. B., Vitulyova E. S., Konshin S. V., Suleimenov I. E.; publ. 26.05.2023.
45. Cheah C. C., Hou S. P., Slotine J. J. E. Region-based shape control for a swarm of robots // Automatica. – 2009. – Vol. 45(10). – P. 2406-2411.
46. Bayındır L. A review of swarm robotics tasks // Neurocomputing. – 2016. – Vol. 172. – P. 292-321.
47. Li C., Wang J., Liu J., Shan J. Cooperative visual–range–inertial navigation for multiple unmanned aerial vehicles // IEEE Transactions on Aerospace and Electronic Systems. – 2023. – Vol. 59(6). – P. 7851-7865.
48. Zhao H., Liu H., Leung Y. W., Chu X. Self-adaptive collective motion of swarm robots // IEEE Transactions on Automation Science and Engineering. – 2018. – Vol. 15(4). – P. 1533-1545.
49. Yahao D., et al. Distributed machine learning for UAV swarms: Computing Sensing, and Semantics // IEEE Internet of Things Journal. – 2024. – Vol. 1(15). – P. 7447-7473.
50. Li C., Guo G., Yi P., Hong Y. Distributed pose-graph optimization with multi-level partitioning for multi-robot SLAM // IEEE Robotics and Automation Letters. – 2024. – Vol. 9(6). – P. 4926-4933.
51. Wei L., Zhijun G. A distributed flocking control strategy for UAV groups // Computer Communications. – 2020. – Vol. 153. – P. 95-101. – DOI: https://doi.org/10.1016/j.comcom.2020.01.076.
52. Xinhua W., Guanyu C., Huajun G., Jiang J. UAV swarm autonomous control based on Internet of Things and artificial intelligence algorithms // Journal of Intelligent & Fuzzy Systems. – 2021. – Vol. 40(4). – P. 7121-7133. – DOI: https://doi.org/10.3233/JIFS-189541.
53. Carli R., Cavone G., Epicoco N., Ferdinando M., Scarabaggio P., Dotoli M. Consensus-Based Algorithms for Controlling Swarms of Unmanned Aerial Vehicles // In Ad-Hoc, Mobile, and Wireless Networks. – 2020. – Vol. 12338.
54. Asaamoning G., Mendes P., Rosário D., Cerqueira E. Drone swarms as networked control systems by integration of networking and computing // Sensors. – 2021. – Vol. 21. – P. 2642. – DOI: https://doi.org/10.3390/s21082642.
55. Elkilany B. G., et al. A proposed decentralized formation control algorithm for robot swarm based on an optimized potential field method // Neural Computing and Applications. – 2021. – Vol. 33. – P. 487-499. – DOI: <https://doi.org/10.1007/s00521-020-05032-0>.
56. Ermukhambetova B., et al. New approaches to the development of information security systems for unmanned vehicles // Indonesian Journal of Electrical Engineering and Computer Science. – 2023. – Vol. 31. – P. 810.
57. Hamamreh J. M., Furqan H. M., Arslan H. Classifications and applications of physical layer security techniques for confidentiality: A comprehensive survey // IEEE Communications Surveys & Tutorials. – 2018. – Vol. 21(2). – P. 1773-1828.
58. Wang D., Bai B., Zhao W., Han Z. A survey of optimization approaches for wireless physical layer security // IEEE Communications Surveys & Tutorials. – 2018. – Vol. 21(2). – P. 1878-1911.
59. Kundu S., Hossain M., Mandal S. Modeling of silicon microring resonator-based programmable logic device for various arithmetic and logic operation in Z-domain // Optical and Quantum Electronics. – 2023. – Vol. 55(2). – P. 175.
60. Beuchat J.-L. Some modular adders and multipliers for field programmable gate arrays // In Proceedings International Parallel and Distributed Processing Symposium. – 2003. – Vol. 8. – DOI: https://doi.org/10.1109/IPDPS.2003.1213353.
61. Abd-Elkader A. A. H., Rashdan M., Hasaneen E. S. A. M. HFA Hamed. Efficient implementation of Montgomery modular multiplier on FPGA // Computers & Electrical Engineering. – 2022. – Vol. 97. – P. 107585. – DOI: https://doi.org/10.1016/j.compeleceng.2021.107585.
62. Kuo C.-T., Wu Y.-C. FPGA implementation of a novel multifunction modulo (2n ± 1) multiplier using radix-4 booth encoding scheme // Applied Sciences. – 2023. – Vol. 13(18). – P. 10407. – DOI: https://doi.org/10.3390/app131810407.
63. Patel B. K., Kanungo J. Efficient Tree Multiplier Design by using Modulo 2^n + 1 Adder // In Emerging Trends in Industry 4.0 (ETI 4.0). – 2021. – P. 1-6. – DOI: https://doi.org/10.1109/ETI4.051663.2021.9619220.
64. Patent EA030205B1. Modulo four adder. Valeriy Pavlovich Suprun. – 2018-07-31.
65. Patent RU2724597C1. Russian Federation, G06F 7/72. Multi-digit parallel adder modulo with serial transfer: 2019144521, 27.12.2019: published 25.06.2020. Petrenko Viacheslav Ivanovich, Stepanian Nerses Ernestovich, Nelidin Iurii Romanovich.

**4 Особенности применения частичных преобразований для вычисления редуцированных числовых преобразований**

**4.1 Аннотация** **четвертой статьи**

Предлагается метод, сводящий вычисление операции сокращенной цифровой свертки к набору сверток, вычисляемых в полях Галуа независимым образом. Под сокращенной цифровой сверткой понимается результат применения операции свертки к функции, принимающей дискретные значения в фиксированном интервале амплитуд, далее сведенной к тому же дискретному масштабу, что и исходная функция. Метод основан на использовании частичных сверток, сводящихся к вычислениям по модулю целого числа $q\_{0}$которое является произведением нескольких простых чисел: $q\_{0}=p\_{1}p\_{2}…p\_{n}$. Показано, что целесообразно использовать разложение числа $q\_{0}$, чтобы $q=p\_{0}p\_{1}p\_{2}…p\_{n}$где $p\_{0}$ дополнительное простое число, для вычисления сокращенной числовой свертки. Это соответствует использованию дополнительных цифр в системе счисления, применяемой для преобразования к частичным сверткам. Обратная процедура, то есть сведение результата вычислений по модулю *q* к результату, соответствующему вычислениям по модулю $q\_{0}$обеспечивается доказанной в данной работе формулой, использующей только целые числа. Обсуждаются возможности практического применения полученных результатов.

**4.2 Введение четвертой статьи**

Операция свертки широко используется в различных науках (оптика [1,2], электроника [3,4] и т.д.).

Цифровая свертка лежит в основе сверточных нейронных сетей (CNN), [5,6]. Под цифровой сверткой в дальнейшем будем понимать свертку, под знаком которой понимаются функции, значения которых соответствуют некоторому конечному набору дискретных величин.

Уместно подчеркнуть, что CNN [5,6] представляют собой одну из наиболее распространенных разновидностей искусственных нейронных сетей (ИНС). Именно с ними можно связать последнюю волну развития ИНС, когда конволюционные нейронные сети смогли показать выдающиеся результаты в области классификации образов [7]. Важной областью применения конволюционных нейронных сетей является распознавание образов и компьютерное зрение [8-10], обработка текстов на естественном языке [11,12], распознавание речи в шумных средах [13] и др.

Как отмечалось в [14,15], для цифровых сигналов существует достаточно широкий простор в выборе функции, служащей моделью сигнала. В частности, в этих работах подчеркивалось, что функции, принимающие значения на множестве вещественных или комплексных чисел, действительно представляют собой некоторый математический объект, позволяющий создать модель сигнала. Выбор такого объекта, позволяющего создать сигнальную модель, в конечном итоге является вопросом согласия и удобства [14,15].

В частности, если рассматривается сигнал, соответствующий дискретным уровням, лежащим в конечном диапазоне амплитуд, то любая функция, которая может служить моделью этого сигнала, должна фактически представлять собой отображение текущей переменной (переменной времени) на некоторое конечное множество. В качестве такого множества могут быть выбраны как поля Галуа [16,17], так и конечные алгебраические кольца [18]. Следует подчеркнуть, что подобные алгебраические структуры в настоящее время все шире используются в информационных технологиях [19-21], причем это относится и к неассоциативным алгебрам [22].

В данной работе мы показываем, что вычисление цифровых сверток можно свести к вычислениям в полях Галуа, в которых, в частности, справедлив численный аналог классической теоремы о свертке [15].

Этот подход представляет интерес, в частности, потому, что от функций, принимающих значения в полях Галуа, допустимо перейти к функциям, значения которых соответствуют переменным многозначной (а также нечеткой) логики [23,24].

Специфические вычисления, выполняемые в полях Галуа или конечных алгебраических кольцах, во многих случаях могут быть сведены к вычислениям по модулю целых чисел. В настоящее время схемы различных типов, позволяющие реализовать такую операцию, разрабатываются достаточно активно [25-28], причем основной областью применения здесь остается криптография. Этот вопрос находит отражение и в патентной литературе [29-32], где предложены различные варианты схем умножителей и сумматоров по модулю некоторых чисел.

Хорошо известно, что вычисления по модулю целых чисел позволяют независимо оперировать компонентами целочисленного представления [33-36]. Это свойство используется, например, для реализации быстрой арифметики [37,38].

Его также можно использовать для представления операции свертки в виде набора частичных сверток, каждая из которых соответствует определенному полю Галуа.

Однако при использовании цифровых сверток есть существенный нюанс. Масштабы дискретизации исходной функции и функции, полученной с помощью оператора свертки, различны.

Чтобы привести их во взаимно однозначное соответствие, необходимо использовать операцию, которую можно интерпретировать как переход к редуцированной цифровой свертке.

Под термином "редуцированная цифровая свертка" мы понимаем результат вычислений операции свертки, приведенный к тому же масштабу дискретизации, что и исходные функции (функции под знаком свертки).

Его можно реализовать с помощью обычной операции округления для дробных чисел, однако представляет интерес убедиться, что при вычислениях цифровая свертка сокращается в терминах операций по модулю целых чисел.

Такой подход позволяет значительно сократить объем вычислений, необходимых для приведения результата операции свертки к тому же масштабу дискретизации, что и масштаб дискретизации функций под своим знаком. При использовании традиционного округления приходится вычислять десятичные дроби. Предлагаемый подход позволяет оперировать только целыми числами. Кроме того, этот подход позволяет реализовать вычисления непосредственно в виде электронных схем, поскольку в настоящее время достаточно хорошо разработаны методы, позволяющие реализовать вычисления в полях Галуа непосредственно в виде электронных схем.

Кроме того, такой подход создает предпосылки для последующего перехода к использованию функций, принимающих значения в переменных многозначной логики и т.д.

Конкретная постановка задачи, решаемой в данной работе, представлена в разделе 2.

Новизна работы заключается в том, что впервые предлагается подход, полностью основанный на вычислениях по модулю целых чисел и позволяющий привести результат вычисления операции свертки к тому же масштабу дискретизации, что и масштаб дискретизации функций под знаком свертки.

Предложенный подход, помимо прочего, позволяет приблизить цифровую свертку к вычислениям в полях Галуа, и в данной работе впервые обосновывается метод выбора простых чисел, позволяющих выполнить эту операцию наиболее простым и удобным способом.

**4.3 Литературный обзор четвертой статьи**

Данная работа лежит в общем русле современных исследований, направленных на повышение эффективности вычислительных систем, в том числе ориентированных на использование конволюционных нейронных сетей.

Одно из важных направлений здесь связано с использованием RNS. Такие системы тесно связаны с быстрой арифметикой, т. е. областью исследований, которой в последние десятилетия уделяется все больше внимания [37-43]. Данная работа решает задачу приведения результата операции свертки к исходному масштабу дискретизации, сохраняя при этом возможность вычислений в терминах РНС, т.е. отвечает указанному выше направлению.

Важно, что последний подход допускает обобщение, поскольку RNS можно рассматривать как частный случай конечных алгебраических колец. Для представления множества, содержащего конечное число элементов, в общем случае могут использоваться самые разнообразные кольца, которые все шире применяются в информационных технологиях [44-47]. Это, в частности, оправдывает использование термина "частичные свертки", который подразумевает возможность последующего перехода к использованию конечных алгебраических колец различных типов. Конкретный пример использования нетривиальных алгебраических колец для цифровой обработки сигналов был приведен в [18]. Обобщение подхода RNS, очевидно, создает дополнительные возможности для цифровой обработки сигналов.

Кроме того, некоторые подмножества алгебраического кольца, используемые для представления множеств, содержащих конечное число элементов, во многих случаях (включая случай, рассматриваемый в данной работе) могут быть поставлены в соответствие полям Галуа. Такие поля также находят все большее применение в информационных технологиях [41-43], причем значительный интерес вызывает использование тесной связи между полями Галуа и РНС [48-53].

Вычисления в таких областях могут быть реализованы непосредственно в виде электронных схем, и исследования в этом направлении также активно ведутся в настоящее время [48-53]. Точнее, вычисления такого рода могут быть реализованы на основе микросхем с перестраиваемой логической структурой, которые также активно используются в настоящее время [54-59]. Поэтому перенос операции свертки (с сохранением исходного масштаба) на дискретизацию представляет интерес, в том числе и с точки зрения реализации соответствующих вычислений (включая реализацию сверточных нейронных сетей) непосредственно с помощью специально запрограммированных чипов.

В качестве примера можно привести вполне определенную проблему, возникающую при обработке спутниковой видеоинформации, предназначенной для дистанционного зондирования Земли. Каналы связи имеют ограниченную пропускную способность, поэтому важно исключить изображения в тех ситуациях, когда поверхность Земли закрыта облаками [60-62]. Для решения этой задачи целесообразно использовать бортовые средства предварительной обработки информации, в качестве которых могут выступать сверточные нейронные сети. Этот конкретный пример демонстрирует важность проблемы, решаемой в данной работе. Достаточно грубый масштаб выборки позволяет исключить кадры с облачностью. Применение операции свертки без приведения к исходному масштабу выборки не только будет соответствовать превышению точности, но и потребует дополнительных вычислительных ресурсов. Преодоление этой трудности становится особенно важным, когда соответствующие операции выполняются непосредственно бортовым компьютером.

Переход к вычислениям в полях Галуа имеет еще один аспект. А именно, в этом случае можно оперировать функциями, принимающими значения на переменных многозначной логики [23,24], [40]. Этот аспект важен в тех случаях, когда, например, мы анализируем связи между предыдущими и последующими значениями временных рядов данных, которые также используются в различных науках. При условии, что система линейна, часто можно предположить, что такие зависимости даются операцией свертки. Переход к использованию многозначных логик в будущем позволяет поставить вопрос о выявлении истинных логических связей в прогнозировании, но сама операция свертки должна быть сведена к логическим переменным. Подчеркнем, что и в этом случае сохранение шкалы дискретизации более чем существенно, поскольку это соответствует использованию одного и того же набора значений логических переменных как для функций под знаком свертки, так и для результата этой операции.

Таким образом, данная работа, предоставляющая возможность свести операцию свертки к вычислениям в полях Галуа относительно небольшого размера (что обеспечивается, в том числе, за счет приведения масштаба дискретизации результата к масштабу дискретизации исходных функций), способствует, в том числе, междисциплинарной интеграции между вышеупомянутыми областями исследований.

**4.4 Постановка проблемы** **четвертой статьи**

Рассмотрим обычное выражение для операции свертки, применяемое к функциям, принимающим дискретные значения

|  |  |
| --- | --- |
| $$U\_{out}\left(i\right)=\sum\_{j}^{}K\left(j\right) U\_{in}\left(i-j\right)$$ | (1) |

где $K\left(j\right)$ - ядро оператора свертки, $U\_{out}\left(i\right)$ и $U\_{in}\left(i\right)$ - функции, которые мы будем рассматривать как "выход" и "вход", соответственно.

Предположим, что функции $U\_{in}\left(i\right)$ и $K\left(j\right)$ являются цифровыми, т.е. каждая из них при каждом из значений индекса принимает одно из $N$ дискретных значений, соответствующих шкале дискретизации, вытекающей из характера решаемой задачи. Будем также считать, что функция $K\left(j\right)$ отлична от нуля только при $M$ тактовые циклы

Тогда количество значений $N\_{1}$которые может принимать функция $U\_{out}\left(i\right)$вообще говоря, лежит в пределах, определяемых неравенством

|  |  |
| --- | --- |
| $$0\leq N\_{1}\leq MN^{2}$$ | (2) |

Видно, что этот числовой диапазон существенно отличается от диапазона изменения исходной функции $U\_{in}\left(i\right)$.

Свести эти два диапазона друг к другу можно с помощью операции деления на подходящее число $M\_{1}$ с последующим округлением до целого числа.

Операции такого рода, однако, не являются алгебраическими. Иначе их нельзя выполнить, используя операции только над целыми числами, например, их нельзя свести к операциям по модулю.

Следовательно, задача состоит в том, чтобы найти алгебраические операции в терминах целых чисел, которые позволили бы обеспечить приведение числовой шкалы функции $U\_{out}\left(i\right)$ приводится к тому же масштабу, что и функция $U\_{in}\left(i\right)$.

Эта проблема решается в данной работе с помощью метода частичных сверток.

**4.5 Методы** **четвертой статьи**

Основным методом, используемым в данной работе, является теория алгебраических колец, в рамках которой доказывается, что существует многообразие колец $R$ которые разлагаются в прямую сумму идеалов $r\_{i}$

|  |  |
| --- | --- |
| $$R=r\_{1}+r\_{2}+…+r\_{n}$$ | (3) |

Каждый из этих идеалов порожден идемпотентными элементами $e\_{i}$

|  |  |
| --- | --- |
| $r\_{i}=Re\_{i}$, | (4) |

которые отменяют друг друга

|  |  |
| --- | --- |
| $e\_{i}e\_{j}=0, i\ne j; e\_{i}e\_{i}=e\_{i}$, | (5) |
|  |  |

и их сумма равна единице кольца $R$

|  |  |
| --- | --- |
| $$\sum\_{i}^{n}e\_{i}=1$$ | (6) |
|  |  |

Простейший пример колец такого типа порождается гомоморфизмом кольца целых чисел на кольцо классов остатков по модулю $p$ для случая, когда число p является произведением нескольких простых чисел $p\_{i}$.

|  |  |
| --- | --- |
| $$P=p\_{1}p\_{2}…p\_{N}$$ | (7) |

В этом случае мы представляем произвольный элемент кольца $R$ в виде

|  |  |
| --- | --- |
| $$u=e\_{1}u\_{1}+e\_{2}u\_{2}+…+e\_{N}u\_{N}$$ | (8) |

где $e\_{i}$ - идемпотентные взаимоисключающие элементы, и $u\_{i}=0,1,2,…,p\_{i}$.

Идемпотентные элементы образуются по правилу

|  |  |
| --- | --- |
| $$e\_{i}=α\_{i}\prod\_{i\ne j}^{N}p\_{j}$$ | (9) |

где $α\_{i}$ - целое число. Выбор этих чисел основан на условии

|  |  |
| --- | --- |
| $$e\_{i}e\_{i}=1$$ | (10) |

Можно посмотреть,

|  |  |
| --- | --- |
| $$e\_{i}p\_{i}≡0 mod P$$ | (11) |

поскольку любое произведение вида (11) содержит множитель $P=p\_{1}p\_{2}…p\_{N}$. Действительно, идемпотентный элемент $e\_{i}$ содержит (9) произведение всех простых множителей числа $P$ кроме $p\_{i}$, и $p\_{i}$ входит в формулу (11) непосредственно.

При выборе целых чисел, $α\_{i}$ также выполняется.

|  |  |
| --- | --- |
| $$e\_{1}+e\_{2}+…+e\_{N}≡1 mod P$$ | (12) |

Отметим, что частным случаем представления (3) является случай, когда мы рассматриваем кольцо остаточных классов по модулю целого числа, имеющего делители нуля. В этом случае представление в виде (3) соответствует вычислениям RNS.

Для иллюстрации рассмотрим пример, соответствующий случаю произведения трех простых чисел 2, 3 и 7. Произведение этих чисел равно 42. Мы рассмотрим кольцо классов остатков по модулю 42.

Идемпотентные взаимоисключающие элементы легко находятся по схеме, соответствующей формуле (9). Составим произведения $3∙7=21$, $3∙2=6$, $2∙7=14$. Все эти элементы будут (при вычислении по модулю 42) аннулировать друг друга, так как результат их умножения друг на друга будет содержать множитель $2∙3∙7$, т.е. этот результат будет кратен 42.

Прямой проверкой можно также убедиться, что для рассматриваемого случая справедливо следующее.

|  |  |
| --- | --- |
| $36∙36≡36 mod 42$, $28∙28≡28 mod 42$, $21∙21≡21mod 42$ | (13) |

Соответственно, при проведении вычислений по модулю 42 запись (8) принимает вид

$u=21∙u\_{1}+28∙u\_{2}+36∙u\_{3}$ (14)

где

$u\_{1}=0,1$, $u\_{2}=0,1,2$, $u\_{3}=0,1,…,6$ (15)

Соотношение (12) также выполняется

$21+28+36≡1 mod 42$ (16)

Отметим также, что представление (8) и его частный случай (14) можно рассматривать как представление чисел, не превышающих 42 в системе счисления с гибридным основанием.

Действительно, можно провести аналогию [63] между обозначением (14) и выражением, определяющим позиционное обозначение числа, скажем, в системе счисления с основанием 10.

$a=a\_{0}+10^{1}∙a\_{1}+10^{2}∙a\_{2}+…$ (17)

Видно, что в обоих случаях существует определенный выбранный набор чисел, который формирует представление любого другого числа в виде последовательности символов из конечного множества (цифр). В случае с нотацией (17) такой набор образован степенями основания числа 10. Но такой выбор, в общем случае, не является обязательным.

Стандартная десятичная нотация $a=…a\_{2}a\_{1}a\_{0}$ Аналогичную систему счисления можно использовать для выражения (8) или его частного случая (14), понимая, что количество символов, соответствующих каждой цифре, не одинаково. Например, для представления (14) его можно записать следующим образом

$u=u\_{1}u\_{2}u\_{3}$ (18)

где позиции, занимаемые конкретными значениями $u\_{i}$ могут быть интерпретированы как аналоги цифр в традиционной системе счисления.

Существенным преимуществом рассматриваемой системы счисления, как известно [37], является возможность независимой операции с аналогами цифр числа.

Операция сложения в десятичном (двоичном и т.д.) представлении, очевидно, связана с тем, что результат, полученный при сложении младших разрядов, вообще говоря, будет влиять на результат, полученный при сложении старших разрядов. При использовании представления вида (14) такая проблема исчезает. Старшие и младшие разряды можно обрабатывать совершенно независимо друг от друга.

Рассмотрим произведение двух чисел, записанных в виде (8)

$u^{\left(1\right)}u^{\left(2\right)}=\left(e\_{1}u\_{1}^{\left(1\right)}+e\_{2}u\_{2}^{\left(1\right)}+…+e\_{N}u\_{N}^{\left(1\right)}\right)\left(e\_{1}u\_{1}^{\left(2\right)}+e\_{2}u\_{2}^{\left(2\right)}+…+e\_{N}u\_{N}^{\left(2\right)}\right)$ (19)

В силу того, что $e\_{i}$ являются взаимоисключающими идемпотентными элементами, мы имеем

$u^{\left(1\right)}u^{\left(2\right)}=e\_{1}u\_{1}^{\left(1\right)}u\_{1}^{\left(2\right)}+e\_{2}u\_{2}^{\left(1\right)}u\_{2}^{\left(2\right)}+…+e\_{N}u\_{N}^{\left(1\right)}u\_{N}^{\left(2\right)}$ (20)

В этом случае вычисление произведений $u\_{i}^{\left(1\right)}u\_{i}^{\left(2\right)}$ фактически выполняется в смысле операции умножения в поле Галуа $GF(p\_{i})$поскольку операция умножения выполняется по модулю $p\_{i}$. Этот вывод справедлив и для операции сложения.

Выражение (20) однозначно показывает, что результат вычисления произведения "старших" цифр действительно полностью независим от результата вычисления произведения "младших" цифр. Более того, такие цифры можно менять местами.

Формулы (19) и (20) позволяют перейти к частичным сверткам.

Существуют условия, при которых можно считать, что формула свертки, применяемая к дискретным функциям (1), записывается в терминах элементов колец, допускающих представление (8). А именно, достаточно выбрать указанные выше простые числа $p\_{i}$ такие, что

$U\_{out}\left(i\right)<p\_{1}p\_{2}…p\_{N}$, (21)

где $N$ - количество цифр в гибридном кодировании.

С учетом сделанного предположения мы можем подставить в формулу (1) разложение по идемпотентным элементам, рассмотренное в предыдущем разделе, и для обоих $K\left(j\right)$, и $U\_{in}\left(i\right)$. Имеем:

$U\_{out}\left(i\right)=\sum\_{j}^{}\left(\sum\_{m}^{}e\_{m}K\_{m}\left(j\right)\right)\left(\sum\_{m}^{}e\_{m}U\_{m,in}\left(i-j\right)\right) $ (22)

Поменяв знаки суммирования, получим

$U\_{out}\left(i\right)=\sum\_{m}^{}e\_{m}\left(\sum\_{j}^{}K\_{m}\left(j\right)U\_{m,in}\left(i-j\right)\right) $ (23)

При выводе формулы (23) учитывается, что идемпотентные элементы $e\_{m}$ аннулируют друг друга при умножении (5). Соответственно, множитель при $e\_{m}$ содержит только значения с одинаковым индексом $m$.

В последней формуле суммирование по индексу $j$ взято в отдельные скобки, чтобы подчеркнуть следующее обстоятельство. Когда функции, соответствующие дискретным сигналам, представлены в гибридной системе счисления, операция свертки факторизуется. С каждой составляющей используемых функций можно работать совершенно независимо.

Соответственно, в этом случае можно говорить о частичных свертках. Имеем:

$U\_{out,m}\left(i\right)=\sum\_{j}^{}K\_{m}\left(j\right)U\_{m,in}\left(i-j\right)$ (24)

Заметим, что операции сложения и умножения в формуле (24), как следует из вышесказанного, выполняются по модулю $p\_{m}$т.е. операция (24) выполняется в поле Галуа $GF\left(p\_{m}\right)$.

В частности, описание любой "оцифрованной" системы (системы, сводимой к дискретным значениям), обладающей свойством инвариантности по отношению к операции сдвига по текущей дискретной переменной, исчерпывается операциями такого рода.

Важно, что использование частичных сверток в терминах полей Галуа допустимо только при выполнении неравенства вида (2). В этом случае максимальное значение, которое может дать операция свертки, не превышает целого числа по модулю, по которому производится вычисление. Следовательно, в этом случае результат операций по модулю совпадает с результатами операций в смысле обычного сложения и умножения целых чисел.

Здесь, однако, возникает нюанс. Действительно, если исходная функция изменялась в диапазоне, соответствующем целым числам, не превышающим $N\_{1}$то результат свертки может изменяться в гораздо более широком диапазоне, формула (2). Следовательно, если исходная функция была отображена на гибридную систему счисления, содержащую $N$ разрядов, то для адекватного представления результата свертки в виде (1) количество разрядов должно быть увеличено.

Решение этой задачи служит основой предлагаемого подхода.

**4.6 Результаты** **четвертой статьи**

Найдем правило, по которому число, представленное в гибридной системе счисления с определенным количеством цифр, может быть записано в аналогичной системе с большим количеством цифр.

Число 1 в произвольном кодировании рассматриваемого типа представляется суммой идемпотентных элементов (пример - формула (16)).

$1≡e\_{1}+e\_{2}+…+e\_{N} mod P$ (25)

где сравнение происходит по модулю целого числа $P$.

Соответственно, при добавлении единицы к любому числу единица добавляется к каждой из цифр. Но для каждой из цифр сложение выполняется по модулю простого числа $p\_{i}$которое соответствует данной цифре. Следовательно,

$u\_{i}\left(n+1\right)≡u\_{i}\left(n\right)+1 \left(p\_{i}\right)$ (26)

Формула (26) позволяет, в частности, явно перейти от записи числа в десятичной системе счисления к записи в системе счисления с гибридным основанием. А именно,

$u\_{i}≡U mod p\_{i}$ (27)

То есть, значение каждой из цифр — это непосредственно исходное число $U$ взятое по модулю $p\_{i}$. Таким образом, формула (27) позволяет свести любую исходную функцию, заданную в обычном числовом виде, к виду, соответствующему гибридной системе счисления.

Значение дополнительного разряда можно также найти по формуле (27).

Дополните гибридную систему чисел, образованную простыми числами $p\_{i}$, $i=1,2,..,N$ еще одной цифрой, соответствующей простому числу $p\_{N+1}$.

Число идемпотентных элементов, входящих в формулу (8), увеличится на единицу, таким образом

$e\_{i}=\tilde{α}\_{i}p\_{N+1}\prod\_{i\ne j}^{N}p\_{j};i=1,..,N$; $e\_{N+1}=\tilde{α}\_{N+1}\prod\_{1}^{N}p\_{j}$ (28)

В частности, это означает, что идемпотентные элементы меняются, но в формуле вида (8) значения компонентов $u\_{i};i=1,..,N$ остаются неизменными. Это следует из формулы (26). Эта формула применима и для вычисления значения дополнительного компонента $u\_{N+1}$.

Отсюда следует, что операции вычисления частичных сверток в полях $GF\left(p\_{i}\right);i=1,..,N$ остаются неизменными при добавлении еще одной цифры. Ее значение, как следует из формулы (26), равно

$u\_{N+1}≡U mod p\_{N+1}$ (29)

Эта формула решает проблему представления чисел при увеличении количества разрядов, однако возникает вопрос, как свести результат вычислений в системе с увеличенным количеством разрядов к исходной шкале дискретизации, в которой рассматриваемые функции принимают целочисленные значения в диапазоне от 0 до $N\_{1}$.

Наиболее очевидный способ приведения к исходной шкале дается формулой

$U\_{out}\left(i\right)=\left⌊\frac{1}{p\_{N+1}}\sum\_{j}^{}K\left(j\right) U\_{in}\left(i-j\right)\right⌋$ (30)

где $\left⌊a\right⌋$ символ функции пола.

Формула получена из следующих соображений. Согласно приведенным выше предположениям, функции под знаком свертки изменяются в диапазоне от 0 до $P$. Результат вычисления свертки изменяется в диапазоне от 0 до $Pp\_{N+1}$. Поэтому, чтобы привести масштаб выборки к исходному, результат вычисления свертки следует разделить на $p\_{N+1}$.

Однако, как отмечалось в разделе 2, актуальной задачей является поиск метода, который позволил бы получить тот же результат, что и формула (30), но только алгебраическим путем, т.е. без использования дробных чисел.

Следующая лемма справедлива.

Для двух натуральных чисел $w\_{1}$ и $w\_{2}$, $w\_{1}>w\_{2}$ и любого натурального n справедливо следующее.

Значение $q$

$q≡\left(w\_{1}-w\_{2}\right)\left⌊\frac{n}{w\_{1}}\right⌋mod w\_{2}$ (31)

где $\left⌊a\right⌋$ символ функции пола, также может быть вычислена как

$q≡\left(n-g\right) mod w\_{2}, g≡n mod w\_{1}$ (32)

Характер изменения величины $q$ с увеличением $n$ для частного случая $p\_{1}=19$, $p\_{2}=17$ иллюстрируется кривой 1, рис. 1. Рисунок подчеркивает, что значение q остается постоянным на каждом из интервалов, соответствующих числу $p\_{2}=17$, то есть меняется скачком, когда значение абсциссы становится кратным 17. Именно это свойство позволяет нам доказать рассматриваемую лемму.



Рисунок 1 – Иллюстрация доказательства леммы; $w\_{1}=19$, $w\_{2}=17$, кривая 1 (синяя) - вычисления $q$ по формуле (31), т.е. через операцию вычисления целой части числа, кривая 2 (красная) - значения суммы $q+3$, где $q$ вычисляется по формуле (32), т.е. с использованием только операций модуляции

На этом рисунке также показана зависимость $q+3$, где q рассчитывается по формуле (32). Зависимость $q$ представлена со сдвигом на 3, чтобы кривые не пересекались. Видно, что если не учитывать искусственный сдвиг на 3, то кривые совпадают. Аналогичные кривые, но для случая $p\_{1}=11$, $p\_{2}=8$ представлены на рис. 2. Этот рисунок также подчеркивает, что значение $q$ меняется скачком, когда значение абсциссы становится кратным $p\_{1}=11$. На этом рисунке также хорошо видно, что конкретное значение $q$ также зависит от значения $p\_{2}$в частности, при выбранных значениях $p\_{1}$ и $p\_{2}$ рассматриваемая функция не имеет ярко выраженного периодического характера. В дальнейшем этот факт будет использован для обоснования выбора соотношения между числами $p\_{1}$ и $p\_{2}$.



Рисунок 2 – Иллюстрация доказательства леммы; $w\_{1}=11$, $w\_{2}=9$, кривая 1 (синяя) - вычисления q по формуле (31), т.е. через операцию вычисления целой части числа, кривая 2 (красная) - значения суммы $q+3$где $q$ вычисляется по формуле (32), то есть с использованием только операций по модулю

Вернемся к доказательству леммы, сформулированной выше.

Существует тождество

$n-n(mod w\_{1})=\left⌊\frac{n}{w\_{1}}\right⌋w\_{1}$ (33)

где обозначение $mod w\_{1}$ записано явно, так как разность в левой части (33) вычисляется в смысле обычных операций с целыми числами.

Тождество (33) следует из того, что произвольное число $n$ может быть представлено в виде суммы остатка от деления на целое число $w\_{1}$ целой части результата такого деления

$n=n\left(mod w\_{1}\right)+\left⌊\frac{n}{w\_{1}}\right⌋w\_{1}$ (34)

При условии, что $w\_{1}>w\_{2}$следующее равенство также верно

$$\left(\left⌊\frac{n}{w\_{1}}\right⌋w\_{1}\right)≡\left(\left(w\_{1}-w\_{2}\right)\left⌊\frac{n}{w\_{1}}\right⌋\right)mod w\_{2}$$

 (35)

Поскольку значение $\left⌊\frac{n}{w\_{1}}\right⌋w\_{2}$ известно, что оно делится на $w\_{2}$.

Следовательно, применяя операцию взятия по модулю $w\_{2}$ к обеим частям равенства (33), получаем

$\left(\left(w\_{1}-w\_{2}\right)\left⌊\frac{n}{w\_{1}}\right⌋\right)≡\left(n-n\left(mod w\_{1}\right)\right) modw\_{2}$ (36)

Лемма доказана.

Самый интересный случай – это когда $w\_{1}-w\_{2}=1$. Тогда левая часть равенства (36) становится равной $\left⌊\frac{n}{w\_{1}}\right⌋\left(mod w\_{2}\right)$.

Заметим, что если мы рассмотрим такой диапазон изменения $n$ что $0<n<w\_{1}w\_{2}$, то $0<\left⌊\frac{n}{w\_{1}}\right⌋<w\_{2}$ имеет место. Следовательно,

$\left⌊\frac{n}{w\_{1}}\right⌋\left(mod w\_{2}\right)=\left⌊\frac{n}{w\_{1}}\right⌋$ (37)

Следовательно, в рассматриваемом случае остаток от деления $n$ на целое число $w\_{1}$ выражается как

$\left⌊\frac{n}{w\_{1}}\right⌋=\left(n-n(mod w\_{1})\right)\left(mod w\_{2}\right)$ (38)

Видно, что в рассматриваемом частном случае доказанная лемма позволяет свести вычисление значения $\left⌊\frac{n}{w\_{1}}\right⌋$ к операциям взятия по модулю. Это позволяет сохранить характер дискретизации шкал "входной" и "выходной" функций.

Получим конкретные формулы, позволяющие выполнить эту операцию.

Начальный диапазон изменения рассматриваемых функций, значения которых представлены в виде (8), ограничен числом $w\_{2}=\prod\_{1}^{N}p\_{j}$. Для выполнения условия, обеспечивающего выполнение формулы (38), простое число $p\_{N+1}$ соответствующее дополнительной цифре, должно быть задано формулой

$p\_{N+1}=\prod\_{1}^{N}p\_{j}+1$ (39)

Примеры простых чисел $p\_{N+1}$ представленных в виде (39), приведены в таблице 2.

Таблица 2.Множители $p\_{i}$ в которой приведены примеры чисел вида (39). (Символ "-" означает, что данный множитель не входит в M)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| $$p\_{4}$$ | $$p\_{3}$$ | $$p\_{2}$$ | $$p\_{1}$$ | $$M=p\_{1}p\_{2}…p\_{s}$$ | $$p\_{N+1}$$ | $$Mp\_{N+1}$$ |
| - | - | 3 | 2 | 6 | 7 | 42 |
| - | - | 5 | 2 | 10 | 11 | 110 |
| - | - | 11 | 2 | 22 | 23 | 506 |
| - | 5 | 3 | 2 | 30 | 31 | 930 |
| - | 7 | 3 | 2 | 42 | 43 | 1 806 |
| - | 11 | 3 | 2 | 66 | 67 | 4 422 |
| - | 7 | 5 | 2 | 70 | 71 | 4 970 |
| - | 17 | 3 | 2 | 102 | 103 | 10 506 |
| - | 13 | 5 | 2 | 130 | 131 | 17 030 |
| - | 19 | 5 | 2 | 190 | 191 | 36 290 |
| 7 | 5 | 3 | 2 | 210 | 211 | 44 310 |
| - | 31 | 5 | 2 | 310 | 311 | 96 410 |
| 11 | 5 | 3 | 2 | 330 | 331 | 109 230 |
| - | 19 | 11 | 2 | 418 | 419 | 175 142 |

Таблица 2 подчеркивает, что даже при использовании относительно небольших чисел $p\_{i}$ и, соответственно, удобных полей $GF(p\_{i})$можно обеспечить вычисление частичных сверток, представляющих практический интерес, поскольку диапазон изменения чисел, представленных в рассматриваемой форме, достаточно велик (он ограничен произведением $Mp\_{N+1}$).

Исходя из вышесказанного, мы рассмотрим случай, когда

$w\_{1}=p\_{N+1}=\prod\_{1}^{N}p\_{j}+1$; $w\_{2}=\prod\_{1}^{N}p\_{j}$ (40)

Чтобы вернуться к исходному масштабу выборки, необходимо применить операцию (38) к результату вычислений с использованием частичных сверток, который представлен в виде.

$\tilde{U}=e\_{1}u\_{1}+e\_{2}u\_{2}+…+e\_{N}u\_{N}+e\_{N+1}u\_{N+1}, mod\left(w\_{1}w\_{2}\right)$ (41)

Это обозначение, в частности, подчеркивает, что целочисленные значения $\tilde{U}$ варьируются в диапазоне $0<\tilde{U}<w\_{1}w\_{2}$.

Согласно формуле (33), для произвольного целого числа $s$ верно, что

$s\left(mod w\_{1}w\_{2}\right)=s-\left⌊\frac{s}{w\_{1}w\_{2}}\right⌋w\_{1}w\_{2}$ (42)

Откуда

$\left(s\left(mod w\_{1}w\_{2}\right)\right)mod w\_{1}≡s mod w\_{1}$, (43)

поскольку $\left⌊\frac{s}{w\_{1}w\_{2}}\right⌋w\_{1}w\_{2}$ делится на $w\_{1}$.

Аналогично,

$\left[s\left(mod w\_{1}w\_{2}\right)\right]\left(mod w\_{2}\right)=s\left(mod w\_{2}\right)$ (44)

В правой части используемой нами формулы (38) находятся только значения, вычисленные по формуле $mod w\_{2}$ или $mod w\_{1}$. Следовательно, как следует из формул (43) и (44), вместо значения $\tilde{U}$ заданного формулой (41), т.е. предусматривающего принятие $mod w\_{1}w\_{2}$можно использовать величину $U$ вычисленное по обычным правилам сложения и умножения

$U=p\_{N+1}\sum\_{i}^{}u\_{i}\tilde{α}\_{i}\prod\_{j\ne 1}^{N}p\_{j}+\tilde{α}\_{N+1}\prod\_{1}^{N}p\_{j}u\_{N+1}$ (45)

Или, принимая во внимание (40)

$U=w\_{1}\left(\sum\_{i}^{}u\_{i}\tilde{α}\_{i}\prod\_{j\ne 1}^{N}p\_{j}\right)+e\_{N+1}u\_{N+1}$ (46)

Применим операцию $mod w\_{1}$ операцию к выражению (32). Тогда

$U\left(mod w\_{1}\right)≡\left(e\_{N+1}u\_{N+1}\right)\left(mod w\_{1}\right)=u\_{N+1}$ (47)

Если будет принято во внимание, что

$e\_{i}≡1 mod p\_{i}$ (48)

Тождество (48) непосредственно следует из (25).

В результате мы получаем следующую расчетную формулу, которая решает поставленную задачу.

$\left⌊\frac{U}{w\_{1}}\right⌋≡\left(\sum\_{i}^{}u\_{i}\tilde{α}\_{i}\prod\_{j\ne 1}^{N}p\_{j}-u\_{N+1}\right) mod w\_{2}$ (49)

Рассмотрим, как именно полученная формула может быть применена для вычисления цифровых сверток.

**4.7 Обсуждение** **четвертой статьи**

Преимущества формулы (49) перед операциями формы (30) заключаются в следующем.

В вычислениях по формуле (49) используются только целые числа. Следовательно, скорость выполнения такого рода операций явно выше, чем скорость выполнения операций, в которых используется операция деления. Более того, все эти операции, в принципе, могут быть реализованы с помощью специальных сумматоров и умножителей по модулю целых чисел. Схемы таких сумматоров и умножителей известны [29-32], и они продолжают совершенствоваться. Следует также учесть, что в настоящее время известны интегральные схемы с конфигурируемой логикой [64,65].

Однако, даже если не принимать во внимание вышеизложенные соображения, известно, что вычисления по формуле (51) имеют преимущество перед операциями типа (30), поскольку диапазон изменения каждой частичной свертки при вычислениях в полях Галуа $GF\left(p\_{i}\right)$ организуется целым числом $p\_{i}$ и, следовательно, все слагаемые, фигурирующие в формуле (51), не будут превышать $w\_{1}=1+\prod\_{1}^{N}p\_{j}$в то время как фактический диапазон вычисления свертки (до приведения к исходному масштабу) составляет $w\_{1}w\_{2}=\left(1+\prod\_{1}^{N}p\_{j}\right)\prod\_{1}^{N}p\_{j}$. Именно это позволяет сохранить исходный масштаб выборки при вычислении свертки.

Рассмотрим конкретный пример, иллюстрирующий это преимущество.

Мы будем использовать кольцо классов остатков по модулю 110.

Элементы такого кольца представляются в виде

$u≡55∙u\_{1}+66∙u\_{2}+100∙u\_{3} mod 110$ (50)

Где

$u\_{1}=0,1$, $u\_{2}=0,1,…,5$, $u\_{3}=0,1,…,10$ (51)

Соотношение (12) также выполняется

$55+66+100≡1 mod 110$ (52)

Простое число 11 можно представить в виде (39): $11=1+2∙5$. Следовательно, формулу (50) можно рассматривать как результат увеличения числа разрядов в системе счисления, соответствующей кольцу классов вычетов по модулю 10. В этом кольце

$u≡5∙u\_{1}+6∙u\_{2} mod 10$ (53)

Где

$u\_{1}=0,1$, $u\_{2}=0,1,…,5$, (54)

Его также можно записать как

$u≡11∙\left(5∙u\_{1}+6∙u\_{2}\right)+100∙u\_{3} mod 110$ (55)

Соответственно, расчетная формула (49) для рассматриваемого конкретного случая принимает следующий вид

$\left⌊\frac{U}{w\_{1}}\right⌋=\left(5∙u\_{1}+6∙u\_{2}-u\_{3}\right) mod 10$ (56)

где $u\_{i}$ значения цифр чисел, полученных в результате вычисления частичных сверток в системе счисления, соответствующей формуле (50).

Применим формулу для вычисления свертки модельных функций.

В качестве первой из таких функций мы будем использовать функцию $f\_{1}$, представленную на рис. 3, кривая 1. В качестве второй модельной функции $f\_{2}$ мы будем использовать функцию прямоугольника, заданную соотношением

$f\_{2}\left(k\right)=\left\{\begin{matrix}0, \left|k\right|>5\\1, \left|k\right|\leq 5\end{matrix}\right.$ (57)

Видно, что $f\_{1}$ моделирует функцию, описывающую некоторый зашумленный переходный процесс. Его свертка с функцией $f\_{2}$ соответствует вычислению скользящего среднего, которое обеспечивает фильтрацию шума. Если допускаются дробные значения, то свертка функций $f\_{1}$ и $f\_{2}$ выражается обычной формулой скользящего среднего

$\left〈f\_{1}\left(k\right)\right〉=\frac{1}{11}\sum\_{i=k-5}^{i=k+5}f\_{1}(k)$ (58)

Формула (60) учитывает, что функция (59) отлична от 0 на 11 тактах, включая такт с индексом 0. Соответственно, суммирование начинается в такт $k-5$ и заканчивается в такт $k+5$.

Расчеты по формуле (58) также представлены на рис. 3, кривая 2. Видно, что результат вычисления свертки действительно дает функцию, описывающую модельный переходный процесс. Функция, полученная в результате применения операции свертки в форме (58), действительно является достаточно гладкой, т.е. эта операция, как и следовало ожидать, подавляет шумы, присутствующие в модельной функции.

Приведенный простой пример, помимо прочего, наглядно показывает, что операция приведения результата свертки к исходному масштабу действительно оправдана. Во многих случаях это диктуется и физическими соображениями.



Рисунок 3 – Модельная функция $f\_{1}$ (кривая 1, красные точки) и результат применения к ней операции расчета, скользящего среднего (кривая 2, зеленые точки)

На рис. 4 также представлена сглаженная кривая модели, полученная из исходной функции методом скользящего среднего (кривая 1).

На этом же рисунке показан результат вычислений по формуле (51) с использованием частичных сверток.

В частности, использовалась формула

$\left〈f\_{1}\right〉\_{0}=\left(5∙\left〈f\_{1}^{1}\right〉+6∙\left〈f\_{1}^{2}\right〉-\left〈f\_{1}^{3}\right〉\right)mod 10$ (59)

Где

$\left〈f\_{1}^{1}\right〉=\left(\sum\_{i=k-5}^{i=k+5}f\_{1}\left(k\right),mod2\right),mod2$ (60)

$\left〈f\_{1}^{2}\right〉=\left(\sum\_{i=k-5}^{i=k+5}f\_{1}\left(k\right),mod5\right),mod5$ (61)

$\left〈f\_{1}^{3}\right〉=\left(\sum\_{i=k-5}^{i=k+5}f\_{1}\left(k\right),mod11\right),mod11$ (62)

Формулы (60) - (62) соответствуют вычислениям в полях Галуа. Значения исходной функции, взятые по модулю соответствующего простого числа $p\_{i}$ суммируются, а затем операция взятия по модулю $p\_{i}$ снова применяется к результату суммирования.

Видно, что представленные кривые совпадают с точностью до округления вниз, т.е. формула (49) действительно дает желаемый результат.

Также видно, что кривая 2 на рис. 4, аппроксимирующая сглаженную кривую с точностью до округления вниз, действительно может быть получена из вычислений, выполненных в полях Галуа.

Подчеркнем, что из рис. 4 видно, что результат вычислений, выполненных только в полях Галуа (в частности, без использования дробных значений), совпадает (с точностью до округления) с результатом, полученным по формуле (58). Таким образом, использованный модельный пример служит наглядной демонстрацией адекватности предложенной методики приведения операции свертки к вычислениям в полях Галуа.

Особенности используемой методики также наглядно иллюстрирует рис. 5, на котором представлены графики результатов промежуточных расчетов по формулам (60) - (62).



Рисунок 4 – Модельная функция $f\_{1}$ сглаженная методом скользящего среднего (кривая 1, зеленые точки) и результат расчета аналога по формуле (51), кривая 2, красные точки

Рисунок 5 – Графики частичных сверток для трех цифр представления числа по модулю 110; рис. а) - в), соответствуют частичным сверткам в полях Галуа $GF(2)$, $GF(5)$ и $GF(11)$ соответственно

Как видно на рис. 5, все функции, представленные на этом рисунке, изменяются в пределах, соответствующих определенному простому числу $p\_{i}$. Кроме того, поведение полученных функций не является упорядоченным, однако использование формулы (51) позволяет достичь желаемого результата. Соответственно, этот рисунок служит еще одной наглядной иллюстрацией сути используемого метода. Результат вышеописанной свертки вычисляется с помощью набора функций, каждая из которых изменяется в конечных пределах, соответствующих определенному полю Галуа. Более того, все операции по вычислению частичных сверток в соответствии с вышеизложенным де-факто выполняются в полях Галуа. Следовательно, для каждой из них применим численный аналог теоремы о свертке [15].

Эти функции являются прямыми аналогами передаточных функций, используемых в классической теории линейных цепей.

Формально мы можем сразу написать

$F\left[U\_{out}\left(i\right)\right]=\left(\sum\_{m}^{}\left[K\_{m}\left(i\right)\right]F\left[U\_{m,in}\left(i\right)\right]\tilde{α}\_{m}\prod\_{j\ne m}^{N}p\_{j}-\left[K\_{m+1}\left(i\right)\right]F\left[U\_{m+1,in}\left(i\right)\right]\right)\left(mod w\_{2}\right)$ (63)

где $F\left[f\left(i\right)\right]$ обозначение преобразования Фурье поля Галуа.

Здесь, конечно, возникает нетривиальная проблема поиска адекватного базиса, представляющего собой аналог гармонических функций, однако уже соотношение (63) показывает, что предложенный подход в перспективе позволит находить передаточные функции для систем различной природы, описываемых операциями цифровой свертки.

Рассмотрим наиболее наглядный пример использования операций, предложенных в данной статье.

Она связана с анализом временных рядов данных, которые возникают в различных научных дисциплинах, например, в метеорологии [66], а также в эконометрике [67] и т.д. Для анализа таких рядов данных (в том числе для целей прогнозирования) часто используются нейронные сети и построенные на их основе интеллектуальные системы различных типов (например, [68-70]). В этом отношении особый интерес представляют, очевидно, конволюционные нейронные сети, поскольку рассматриваемые системы, как правило, обладают свойством инвариантности по отношению к операции сдвига по времени. Предлагаемый подход позволяет свести любые операции свертки к операциям в терминах логических переменных, что, в частности, представляет интерес с точки зрения выявления истинных причинно-следственных связей, присущих конкретной системе.

Однако этот пример приведен лишь в качестве иллюстрации. Гораздо более важным представляется следующий шаг, связанный с применением теоремы о численной свертке [18]. А именно, если все операции, соответствующие свертке в исходный масштаб дискретизации, свести к сверткам в смысле полей Галуа, то от свертки допустимо перейти к аналогу передаточной функции (образ Фурье-Галуа свертки есть произведение образов Фурье-Галуа функций под своим знаком). В перспективе это создает предпосылки для формирования сверточных нейронных сетей с заранее заданными свойствами (т.е. нейронных сетей, обладающих заданным аналогом передаточной функции). Этот шаг, однако, требует формирования соответствующих наборов ортогональных базисных функций. Уместно подчеркнуть, что интерес к практическому использованию многозначных логик в настоящее время связан не только с совершенствованием цифровой обработки сигналов, но и с совершенствованием ИИ. В частности, как показано в [71], биологический прототип ИИ - человеческий интеллект - не сводим к двоичной логике. В то же время не вызывает сомнений, что обработка сигналов человеческим мозгом также соответствует вышеупомянутому свойству симметрии. Это открывает дополнительные перспективы для использования результатов, полученных в данной работе.

**4.8 Выводы** **четвертой статьи**

Таким образом, в работе предложен специальный метод, позволяющий привести результат операции свертки к тому же масштабу дискретизации, что и исходные функции (функции под знаком свертки).

В этом случае число дискретных уровней функций под знаком свертки совпадает с числом дискретных уровней, соответствующих результату операции свертки.

Операцию свертки, при которой сохраняется характер дискретизации исходной и результирующей функций, имеет смысл рассматривать как редуцированную свертку.

Предложенный подход позволяет перенести операцию сокращенного сверточного исчисления на вычисления в полях Галуа. Конструктивность этого подхода демонстрируется на конкретном модельном примере.

При построении этого метода вычисления на основе RNS рассматриваются как частный случай вычислений в конечных алгебраических кольцах, что создает предпосылки для его использования при переходе к алгебраическим кольцам различных многообразий.

Дальнейшее развитие этого подхода предполагает, в частности, использование функций, принимающих значения, соответствующие переменным многозначной логики.

Полученные результаты также представляют интерес с точки зрения создания конволюционных нейронных сетей с заданными свойствами.

**4.9 Список литературы четвертой статьи**

1. Goodman J.W. Introduction to Fourier Optics. – 3rd ed. – Englewood, CO, USA: Roberts & Co. – 2005. – 491 p.
2. Tyson R.K. Principles and Applications of Fourier Optics. – Bristol, UK: IOP Publishing. – 2014. – 117 p.
3. Darlington S. A History of Network Synthesis and Filter Theory for Circuits Composed of Resistors, Inductors, and Capacitors // IEEE Transactions on Circuits and Systems. – 1984. – Vol. 31. – P. 3-13. – DOI: https://doi.org/10.1109/TCS.1984.1085415.
4. Izadian A. Fundamentals of Modern Electric Circuit Analysis and Filter Synthesis: A Transfer Function Approach. – Cham, Switzerland: Springer International Publishing. – 2019. – 115 p. – DOI: https://doi.org/10.1007/978-3-030-02484-0.
5. Gu J., Wang Z., Kuen J., Ma L., Shahroudy A., Shuai B., Liu T., Wang X., Wang G., Cai J., et al. Recent Advances in Convolutional Neural Networks // Pattern Recognition. – 2018. – Vol. 77. – P. 354-377. – DOI: https://doi.org/10.1016/j.patcog.2017.10.013.
6. Aghdam H.H., Heravi J.E. Guide to Convolutional Neural Networks. – Cham, Switzerland: Springer International Publishing. – 2017. – 282 p. – DOI: https://doi.org/10.1007/978-3-319-57550-6.
7. Krizhevsky A., Sutskever I., Hinton G.E. Imagenet classification with deep convolutional neural networks // Advances in Neural Information Processing Systems. – 2012. – Vol. 25. – P. 1097-1105.
8. Monteiro A., Oliveira M., Oliveira R., Silva T. Embedded Application of Convolutional Neural Networks on Raspberry Pi for SHM // Electronics Letters. – 2018. – Vol. 54. – P. 680-682. – DOI: https://doi.org/10.1049/el.2018.0877.
9. Shichijo S., Nomura S., Aoyama K., Nishikawa Y., Miura M., Shinagawa T., Takiyama H., Tanimoto T., Ishihara S., Matsuo K., et al. Application of Convolutional Neural Networks in the Diagnosis of Helicobacter Pylori Infection Based on Endoscopic Images // EBioMedicine. – 2017. – Vol. 25. – P. 106-111. – DOI: https://doi.org/10.1016/j.ebiom.2017.10.014.
10. Howard A.G., Zhu M., Chen B., Kalenichenko D., Wang W., Weyand T., Andreetto M., Adam H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications // arXiv. – 2017. – arXiv:1704.04861. – DOI: https://doi.org/10.48550/arxiv.1704.04861.
11. Ketkar N., Moolayil J. Convolutional Neural Networks // In Deep Learning with Python. – Berkeley, CA, USA: Apress. – 2021. – P. 197-242. – DOI: https://doi.org/10.1007/978-1-4842-5364-9\_6.
12. Zhang C., Zhao H., Cao M. Research on General Text Classification Model Integrating Character-Level Attention and Multi-Scale Features // In Proceedings of the 2021 10th International Conference on Computing and Pattern Recognition. – Shanghai, China, 15–17 October 2021. – P. 183–187. – DOI: https://doi.org/10.1145/3497623.3497652.
13. McLaren M., Lei Y., Scheffer N., Ferrer L. Application of Convolutional Neural Networks to Speaker Recognition in Noisy Conditions // In Proceedings of the Fifteenth Annual Conference of the International Speech Communication Association, Interspeech 2014. – Singapore, 14–18 September 2014. – P. 686–690. – DOI: https://doi.org/10.21437/Interspeech.2014-172.
14. Moldakhan I., Matrassulova D.K., Shaltykova D.B., Suleimenov I.E. Some Advantages of Non-Binary Galois Fields for Digital Signal Processing // Indonesian Journal of Electrical Engineering and Computer Science. – 2021. – Vol. 23. – P. 871. – DOI: https://doi.org/10.11591/ijeecs.v23.i2.pp871-878.
15. Vitulyova E.S., Matrassulova D.K., Suleimenov I.E. New Application of Non-Binary Galois Fields Fourier Transform: Digital Analog of Convolution Theorem // Indonesian Journal of Electrical Engineering and Computer Science. – 2021. – Vol. 23. – P. 1718. – DOI: https://doi.org/10.11591/ijeecs.v23.i3.pp1718-1726.
16. Suleimenov I.E., Vitulyova Y.S., Matrassulova D.K. Features of Digital Signal Processing Algorithms Using Galois Fields GF(2^n+1) // PLoS ONE. – 2023. – Vol. 18. – P. e0293294. – DOI: https://doi.org/10.1371/journal.pone.0293294.
17. Vitulyova E.S., Matrassulova D.K., Suleimenov I.E. Construction of Generalized Rademacher Functions in Terms of Ternary Logic: Solving the Problem of Visibility of Using Galois Fields for Digital Signal Processing // International Journal of Electronics and Telecommunications. – 2022. – P. 237–244.
18. Matrassulova D.K., Vitulyova Y.S., Konshin S.V., Suleimenov I.E. Algebraic Fields and Rings as a Digital Signal Processing Tool // Indonesian Journal of Electrical Engineering and Computer Science. – 2022. – Vol. 29. – P. 206. – DOI: https://doi.org/10.11591/ijeecs.v29.i1.pp206-216.
19. Kuo Y.-M., Garcia-Herrero F., Ruano O., Maestro J.A. RISC-V Galois Field ISA Extension for Non-Binary Error-Correction Codes and Classical and Post-Quantum Cryptography // IEEE Transactions on Computers. – 2022. – Vol. 72. – P. 682–692. – DOI: https://doi.org/10.1109/TC.2022.3174587.
20. Nazarov L.E. Investigation of Noise Immunity of Optimal Symbol Reception of Frequency-Efficient Signals with Correction Coding in Non-Binary Galois Fields // Journal of Communications Technology and Electronics. – 2023. – Vol. 68. – P. 960–965. – DOI: https://doi.org/10.1134/S106422692309019X.
21. Jagadeesh H., Joshi R., Rao M. Group Secret-Key Generation Using Algebraic Rings in Wireless Networks // IEEE Transactions on Vehicular Technology. – 2021. – Vol. 70. – P. 1538–1553. – DOI: https://doi.org/10.1109/TVT.2021.3054031.
22. Alcayde A., Ventura J., Montoya F.G. Hypercomplex Techniques in Signal and Image Processing Using Network Graph Theory: Identifying Core Research Directions [Hypercomplex Signal and Image Processing] // IEEE Signal Processing Magazine. – 2024. – Vol. 41. – P. 14–28. – DOI: https://doi.org/10.1109/MSP.2024.3365463.
23. Suleimenov I.E., Vitulyova Y.S., Kabdushev S.B., Bakirov A.S. Improving the Efficiency of Using Multivalued Logic Tools // Scientific Reports. – 2023. – Vol. 13. – P. 1108. – DOI: https://doi.org/10.1038/s41598-023-28272-1.
24. Suleimenov I.E., Vitulyova Y.S., Kabdushev S.B., Bakirov A.S. Improving the Efficiency of Using Multivalued Logic Tools: Application of Algebraic Rings // Scientific Reports. – 2023. – Vol. 13. – P. 22021. – DOI: https://doi.org/10.1038/s41598-023-49593-1.
25. Tynymbayev S., Berdibayev R.S., Omar T., Gnatyuk S.A., Namazbayev T.A., Adilbekkyzy S. Devices for Multiplying Modulo Numbers with Analysis of the Lower Bits of the Multiplier // Bulletin of the National Academy of Sciences of the Republic of Kazakhstan. – 2019. – Vol. 4. – P. 38–45. – DOI: <https://doi.org/10.32014/2019.2518-1467.90>.
26. Sayed-Ahmed A., Große D., Kühne U., Soeken M., Drechsler R. Formal verification of integer multipliers by combining Gröbner basis with logic reduction // In Proceedings of the 2016 Design, Automation & Test in Europe Conference & Exhibition (DATE). – Dresden, Germany, 14–18 March 2016. – P. 1048–1053.
27. Munoz-Coreas E., Thapliyal H. Quantum Circuit Design of a T-Count Optimized Integer Multiplier // IEEE Transactions on Computers. – 2019. – Vol. 68. – P. 729–739. – DOI: https://doi.org/10.1109/TC.2018.2882774.
28. Sousa L., Antao S., Martins P. Combining Residue Arithmetic to Design Efficient Cryptographic Circuits and Systems // IEEE Circuits and Systems Magazine. – 2016. – Vol. 16. – P. 6–32. – DOI: https://doi.org/10.1109/MCAS.2016.2614714.
29. Hidenori E., Kiyoto K., Denki C. Circuit for Modulo Multiplication and Exponentiation Arithmetic // Patent EP0801345B1. – 16 October 2002.
30. Lablans P. Multi-Value Digital Calculating Circuits, Including Multipliers // Patent US20060031278A1. – 9 February 2006.
31. Irkhin V.P., Obukhov A.N., Gul’bin S.S. G06F 7/49. Device for Modulo Addition and Subtraction of Numbers // Patent RU2145112C1. – 27 January 2000.
32. Pisek E., Henige T.M. Method and Apparatus for Efficient Modulo Multiplication // Patent US8417756B2. – 2003.
33. Nakahara H., Sasao T. A Deep Convolutional Neural Network Based on Nested Residue Number System // In Proceedings of the 2015 25th International Conference on Field Programmable Logic and Applications (FPL). – London, UK, 2–4 September 2015. – P. 1–6. – DOI: https://doi.org/10.1109/FPL.2015.7293933.
34. Valueva M.V., Nagornov N.N., Lyakhov P.A., Valuev G.V., Chervyakov N.I. Application of the Residue Number System to Reduce Hardware Costs of the Convolutional Neural Network Implementation // Mathematics and Computers in Simulation. – 2020. – Vol. 177. – P. 232–243. – DOI: https://doi.org/10.1016/j.matcom.2020.04.031.
35. Jenkins W., Leon B. The Use of Residue Number Systems in the Design of Finite Impulse Response Digital Filters // IEEE Transactions on Circuits and Systems. – 1977. – Vol. 24. – P. 191–201. – DOI: https://doi.org/10.1109/TCS.1977.1084321.
36. Aithal G., Bhat K.N.H., Sripathi U. Implementation of Stream Cipher System Based on Representation of Integers in Residue Number System // In Proceedings of the 2010 IEEE 2nd International Advance Computing Conference (IACC). – Patiala, India, 19–20 February 2010. – P. 210–217. – DOI: https://doi.org/10.1109/IADCC.2010.5423007.
37. Dubey A., Ahmad A., Pasha M.A., Cammarota R., Aysu A. ModuloNET: Neural Networks Meet Modular Arithmetic for Efficient Hardware Masking // ACR Transactions on Cryptographic Hardware and Embedded Systems. – 2021. – P. 506–556. – DOI: https://doi.org/10.46586/tches.v2022.i1.506-556.
38. Yassine H.M. Fast Arithmetic Based on Residue Number System Architectures // In Proceedings of the 1991 IEEE International Symposium on Circuits and Systems (ISCAS). – Singapore, 11–14 June 1991. – Vol. 5. – P. 2947–2950. – DOI: https://doi.org/10.1109/ISCAS.1991.176163.
39. Bos J.W., Friedberger S. Fast Arithmetic Modulo 2x py ± 1 // In Proceedings of the 2017 IEEE 24th Symposium on Computer Arithmetic (ARITH). – London, UK, 24–26 July 2017. – P. 148–155.
40. Torabi Z., Jaberipur G., Belghadr A. Fast Division in the Residue Number System {2n + 1, 2n, 2n-1} Based on Shortcut Mixed Radix Conversion // Computers & Electrical Engineering. – 2020. – Vol. 83. – P. 106571. – DOI: <https://doi.org/10.1016/j.compeleceng.2020.106571>.
41. Markov V.T., Mikhalev A.V., Nechaev A.A. Nonassociative Algebraic Structures in Cryptography and Coding // Journal of Mathematical Sciences. – 2020. – Vol. 245. – P. 178–196. – DOI: https://doi.org/10.1007/s10958-020-04685-5.
42. Sanam N., Ali A., Shah T., Farooq G. Non-Associative Algebra Redesigning Block Cipher with Color Image Encryption // Computers, Materials & Continua. – 2021. – Vol. 67. – P. 1–21. – DOI: https://doi.org/10.32604/cmc.2021.014442.
43. Liu P., Pan Z., Lei J. Parameter Identification of Reed-Solomon Codes Based on Probability Statistics and Galois Field Fourier Transform // IEEE Access. – 2019. – Vol. 7. – P. 33619–33630. – DOI: https://doi.org/10.1109/ACCESS.2019.2904718.
44. Huang Q., Tang L., He S., Xiong Z., Wang Z. Low-Complexity Encoding of Quasi-Cyclic Codes Based on Galois Fourier Transform // IEEE Transactions on Communications. – 2014. – Vol. 62. – P. 1757–1767. – DOI: https://doi.org/10.1109/TCOMM.2014.2316174.
45. Hazzazi M.M., Attuluri S., Bassfar Z., Joshi K. A Novel Cipher-Based Data Encryption with Galois Field Theory // Sensors. – 2023. – Vol. 23. – P. 3287. – DOI: https://doi.org/10.3390/s23063287.
46. Asif M., Asamoah J.K.K., Hazzazi M.M., Alharbi A.R., Ashraf M.U., Alghamdi A.M. A Novel Image Encryption Technique Based on Cyclic Codes over Galois Field // Computational Intelligence and Neuroscience. – 2022. – Vol. 2022. – P. 1912603. – DOI: https://doi.org/10.1155/2022/1912603.
47. Hiasat A., Sousa L. On the Design of RNS Inter-Modulo Processing Units for the Arithmetic-Friendly Moduli Sets {2n + k, 2n − 1, 2n + 1 − 1} // The Computer Journal. – 2019. – Vol. 62. – P. 292–300. – DOI: https://doi.org/10.1093/comjnl/bxy119.
48. Ha J., Kim S., Choi W., Lee J., Moon D., Yoon H., Cho J. Masta: An HE-Friendly Cipher Using Modular Arithmetic // IEEE Access. – 2020. – Vol. 8. – P. 194741–194751. – DOI: https://doi.org/10.1109/ACCESS.2020.3033564.
49. Hu K., Zhang D., Xia M. CDUNet: Cloud Detection UNet for Remote Sensing Imagery // Remote Sensing. – 2021. – Vol. 13. – P. 4533. – DOI: https://doi.org/10.3390/rs13224533.
50. Li Z., Shen H., Weng Q., Zhang Y., Dou P., Zhang L. Cloud and Cloud Shadow Detection for Optical Satellite Imagery: Features, Algorithms, Validation, and Prospects // ISPRS Journal of Photogrammetry and Remote Sensing. – 2022. – Vol. 188. – P. 89–108. – DOI: <https://doi.org/10.1016/j.isprsjprs.2022.03.020>.
51. Suleimenov I., Vitulyova Y., Bakirov A. Hybrid Number Systems: Application for Calculations in Galois Fields // In Proceedings of the 2022 3rd Asia Conference on Computers and Communications (ACCC). – Shanghai, China, 16–18 December 2022. – P. 126–130. – DOI: https://doi.org/10.1109/ACCC58361.2022.00028.
52. Sadeghi A., Shiri N., Rafiee M., Rahimi P. A Low-Power Pseudo-Dynamic Full Adder Cell for Image Addition // Computers & Electrical Engineering. – 2020. – Vol. 87. – P. 106787. – DOI: https://doi.org/10.1016/j.compeleceng.2020.106787.
53. Özkilbaç B. Implementation and Design of 32 Bit Floating-Point ALU on a Hybrid FPGA-ARM Platform // Journal of Brilliant Engineering. – 2019. – Vol. 1. – P. 26–32. – DOI: https://doi.org/10.36937/ben.2020.001.005.
54. Yan B., Chan P.W., Li Q., He Y., Shu Z. Dynamic Analysis of Meteorological Time Series in Hong Kong: A Nonlinear Perspective // International Journal of Climatology. – 2021. – Vol. 41. – P. 4920–4932. – DOI: https://doi.org/10.1002/joc.7106.
55. Nonejad N. An overview of dynamic model averaging techniques in time-series econometrics // Journal of Economic Surveys. – 2021. – Vol. 35. – P. 566–614. – DOI: https://doi.org/10.1111/joes.12410.
56. Kumar G., Singh U.P., Jain S. Hybrid Evolutionary Intelligent System and Hybrid Time Series Econometric Model for Stock Price Forecasting // International Journal of Intelligent Systems. – 2021. – Vol. 36. – P. 4902–4935. – DOI: https://doi.org/10.1002/int.22495.
57. Chatterjee A., Bhowmick H., Sen J. Stock Price Prediction Using Time Series, Econometric, Machine Learning, and Deep Learning Models // In Proceedings of the 2021 IEEE Mysore Sub Section International Conference (MysuruCon). – Mysuru, India, 16–17 October 2021. – P. 289–296. – DOI: https://doi.org/10.1109/MysuruCon52639.2021.9641610.
58. Rahman M., Shakeri M., Khatun F., Tiong S.K., Alkahtani A.A., Samsudin N.A., Amin N., Pasupuleti J., Hasan M.K. A Comprehensive Study and Performance Analysis of Deep Neural Network-Based Approaches in Wind Time-Series Forecasting // Journal of Reliable Intelligent Environments. – 2023. – Vol. 9. – P. 183–200. – DOI: https://doi.org/10.1007/s40860-021-00166-x.
59. Gabrielyan O.A., Vitulyova E., Suleimenov I.E. Multi-valued logics as an advanced basis for artificial intelligence // Wisdom. – 2022. – Vol. 1. – P. 170–181. – DOI: <https://doi.org/10.24231/wisdom.v21i1.721>.

**ЗАКЛЮЧЕНИЕ**

Таким образом, в данной работе приводятся русские тексты публикаций, представленных на соискание степени доктора философии (PHD), соответствующие п. 5-1 Правил присуждения степеней.