УДК 681.31

Я.Н. НИКОЛАЙЧУК, П.В. ГУМЕННЫЙ

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ, МЕТОДЫ И ПРОЦЕССОРЫ ПРЕОБРАЗОВАНИЯ ИНФОРМАЦИИ В КОДАХ ПОЛЯ ГАЛУА НА БАЗЕ ВЕРТИКАЛЬНО-ИНФОРМАЦИОННОЙ ТЕХНОЛОГИИ

Аннотация. Рассмотрен метод выполнения арифметической операции сложения и умножения в базисе Галуа. Разработаны базовая структура сумматора, в котором увеличено быстродействие выполнения операций в кодах Галуа, и функциональная структура спецпроцессора на основе вертикально-информационной технологии.

Ключевые слова: теоретико-числовой базис (ТЧБ), вертикально-информационная технология (ВИТ), спецпроцессор.

ПОСТАНОВКА ПРОБЛЕМЫ

Методологической основой большинства современных процессоров известных фирм — производителей средств вычислительной техники является теоретико-числовой базис (ТЧБ) Радемахера, который порождает двоичную систему исчисления. Исследование тенденций развития вычислительной техники свидетельствует о теоретическом исчерпании вычислительных ресурсов базиса Радемахера для построения процессоров, к которым предъявляются все более жесткие требования относительно габаритов, увеличения разрядности (более 1024 бит), быстродействия, архитектуры и минимизации аппаратной сложности. Мировой опыт за последние годы демонстрирует тенденции к исследованию и успешному применению других базисов: унитарного, Хаара, Крейга, Крестенсона и Галуа, которые также порождают системы исчисления [1]. Достаточно актуальным для разработки универсальных процессоров и спецпроцессоров является базис Галуа, математические преобразования и схемотехника компонентов вычислительных средств которых хорошо согласовываются с вертикально-информационной технологией (ВИТ) [2]. Основным свойством кодовых систем Галуа является вертикальная логическая рекурсивная взаимозависимость кодовых элементов, что позволяет осуществить кодирование каждого элемента параллельных кодов последовательным бит-ориентированным кодом Галуа. Система кодирования Галуа, имеющая рекурсивный порядок кодовых элементов, владеет одним из лучших показателей компактности кодирования информации [3], что позволяет проектировать спецпроцессоры с минимальным количеством коммуникационных соединений в виде бит-ориентированных шин данных, адреса и управления.

АНАЛИЗ ИССЛЕДОВАНИЙ И ПУБЛИКАЦИЙ

Известны успешные разработки ТЧБ Галуа для построения спецпроцессоров и их компонентов, выполняющих обработку бит-ориентированных потоков, выполненных известными зарубежными фирмами — разработчиками микропроцессорной техники и электроники:

— корпорацией Fujitsu Limited [4] разработан процессор, арифметико-логическое устройство которого обеспечивает простую реализацию арифметических и логических операций в базисе Галуа и снижает аппаратную сложность процессора по сравнению с аналогичным процессором в базисе Радемахера;

— специалистами Sony Corporation [5] разработана комбинационная схема, которая используется для коррекции ошибок при передаче и записи цифровой информации;

— корпорация Analog Devices разработала устройство [6] для умножения, умножения–сложения/умножения–накопления, которое характеризуется более высоким быстродействием по сравнению с базисом Радемахера;

— корпорация Matsushita Electric Industrial разработала устройство [7] для проверки/исправления однократной ошибки, которое используется при записи/воспроизведении данных на оптический диск;

— Тернопольским национальным экономическим университетом [8, 9] разработаны фундаментальные теоретические положения арифметики вычислительных операций, основанные на теории кодов поля Галуа, а также представлены эффективные решения прикладных задач формирования, передачи и цифровой обработки информационных потоков на основе спецпроцессоров, которые реализуют преобразование полей Галуа и базовые положения вертикально-информационной технологии.

Спецпроцессоры и их компоненты в базисе Галуа нашли широкое применение в отраслях цифровой обработки сигналов, компьютерных сетях и проблемно ориентированных компьютеризованных системах. Существенно расширили информационные технологии и теорию построения компонентов процессоров на основе вертикально-информационной технологии и мультибазисных процессоров зарубежные и отечественные ученые [2, 10–15]. Однако при построении спецпроцессоров недостаточно использованы возможности теории выполнения арифметико-логических операций в базисе Галуа и практически не освоены в производстве базовые компоненты процессоров ВИТ. Такими компонентами являются АЦП сканирующего типа с выходными кодами Галуа [16], арифметико-логические устройства бит-ориентированной обработки данных [8], вертикальные адресные дешифраторы [9] и другие. Перспективу также определяют возможности построения высокопроизводительных мультибазисных RCG-процессоров [17].

Цель настоящей статьи — разработка теоретических принципов и методов преобразования информации на основе ВИТ-технологии с использованием теории кодов поля Галуа и реализация функциональной структуры спецпроцессора ВИТ.

ТЕОРЕТИЧЕСКИЕ ПРИНЦИПЫ КОДОВЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ ГАЛУА И ПРИНЦИПЫ ВЫПОЛНЕНИЯ АРИФМЕТИЧЕСКИХ ОПЕРАЦИЙ СЛОЖЕНИЯ И УМНОЖЕНИЯ

Важная особенность представления данных в базисе Галуа — свойство рекуррентности [2]. Суть рекуррентности состоит в максимальной упаковке бит-ориентированной последовательности согласно выражению

$$C_{(x)} = \sum_{i=0}^{n-1} c_i x_i \bmod P, \tag{1}$$

при этом

$$\begin{aligned} C_{(x)} = {} & (a_{n-1}d_{n-1}^{n-1} + a_{n-2}d_{n-2}^{n-1} + \ldots + a_1 d_1^{n-1} + a_0 d_0^{n-1})x^{n-1} \bmod P + \\ & + (a_{n-1}d_{n-1}^{n-2} + a_{n-2}d_{n-2}^{n-2} + \ldots + a_1 d_1^{n-2} + a_0 d_0^{n-2})x^{n-2} \bmod P + \ldots \\ & \ldots + (a_{n-1}d_{n-1}^{1} + a_{n-2}d_{n-2}^{1} + \ldots + a_1 d_1^{1} + a_0 d_0^{1})x \bmod P + \\ & + (a_{n-1}d_{n-1}^{0} + a_{n-2}d_{n-2}^{0} + \ldots + a_1 d_1^{0} + a_0 d_0^{0}) \bmod P. \end{aligned}$$

В частном случае для простейшего примитивного полинома выражение (1) имеет вид

$$X_{i+1} = \sum_{i=1}^{n} (X_i \oplus X_{i-j}), \qquad (2)$$

где $X_i \in \overline{0,1}$, $i \le j \le n$, знак $\oplus$ представляет символ сложения по $\bmod 2$, n — число пар элементов кодового ключа.

Наиболее удобной формой представления кодов поля Галуа является выражение примитивного полинома [2]

$$G(x) = a_{n-1} \cdot x^{n-1} + a_{n-2} \cdot x^{n-2} + \ldots + a_1 \cdot x + a_0,$$

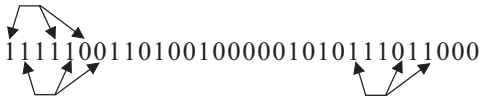
где $a_0 \div a_{n-1}$ — двоичные переменные 0 или 1, которые определяют соответствующие значения разрядов кодовых комбинаций. Например, в поле Галуа $GF\begin{pmatrix}5\\2\end{pmatrix}$ с ключом 10010 на основе полинома $x^5 + x^2 + 1$ формируется последовательность элементов $a_0, a_1, a_2, \ldots, a_{30}$, которая кодирует числа в диапазоне $0, 1, 2, \ldots, 30$ и имеет вид

$$11111001101001000010101110110 00,$$

где $(11111) = (b_5, b_4, b_3, b_2, b_1)$ — стартовая последовательность кода Галуа, которая позволяет рекуррентно представить числа логическими выражениями:

$$b_5, b_4, b_3, b_2, b_1,\ b_2 \oplus b_5,\ b_1 \oplus b_4,\ b_2 \oplus b_3 \oplus b_5,\ b_1 \oplus b_2 \oplus b_4, b_1 \oplus b_2 \oplus b_3 \oplus b_5,$$
$$b_1 \oplus b_4 \oplus b_5,\ b_2 \oplus b_3 \oplus b_4 \oplus b_5,\ b_1 \oplus b_2 \oplus b_3 \oplus b_4,\ b_1 \oplus b_3 \oplus b_5,\ b_4 \oplus b_5,$$
$$b_3 \oplus b_4, b_2 \oplus b_3, b_1 \oplus b_2, b_1 \oplus b_2 \oplus b_5, b_1 \oplus b_2 \oplus b_4 \oplus b_5, b_1 \oplus b_2 \oplus b_3 \oplus b_4 \oplus b_5,$$
$$b_1 \oplus b_3 \oplus b_4 \oplus b_5,\ b_3 \oplus b_4 \oplus b_5,\ b_2 \oplus b_3 \oplus b_4,\ b_1 \oplus b_2 \oplus b_3,$$
$$b_1 \oplus b_5,\ b_2 \oplus b_4 \oplus b_5,\ b_1 \oplus b_3 \oplus b_4, b_3 \oplus b_5,\ b_2 \oplus b_4,\ b_1 \oplus b_3.$$

Ниже отражен принцип формирования 5-разрядного кода Галуа согласно выражению $G_{i+1} = G_i \oplus G_{i-n}$:



Аналогично на основе других примитивных полиномов формируются (2^{n-1})-разрядные двоичные коды Галуа. Векторное представление элементов поля $GF\begin{pmatrix}5\\2\end{pmatrix}$ аппаратно реализуется структурной схемой, приведенной на рис. 1.

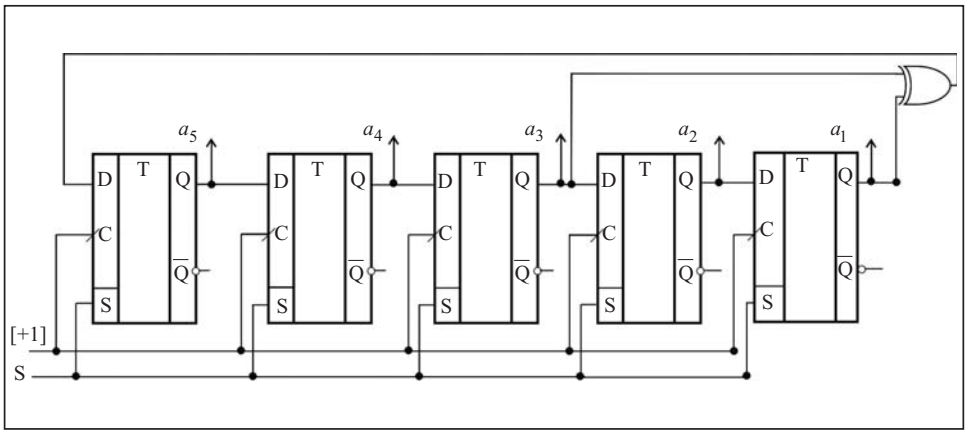


Рис. 1. Структурная схема векторного представления элементов поля Галуа $GF\begin{pmatrix}5\\2\end{pmatrix}$

Таблица 1. Логическое представление 5-разрядных кодов Галуа

Номер кода	Код Галуа	Разряды кодов Галуа, выраженные через b_1, b_2, b_3, b_4, b_5				
0	11111	b_5	b_4	b_3	b_2	b_1
1	11110	b_4	b_3	b_2	b_1	$b_2 \oplus b_5$
2	11100	b_3	b_2	b_1	$b_2 \oplus b_5$	$b_1 \oplus b_4$
3	11001	b_2	b_1	$b_2 \oplus b_5$	$b_1 \oplus b_4$	$b_2 \oplus b_3 \oplus b_5$
4	10011	b_1	$b_2 \oplus b_5$	$b_1 \oplus b_4$	$b_2 \oplus b_3 \oplus b_5$	$b_1 \oplus b_2 \oplus b_4$
5	00110	$b_2 \oplus b_5$	$b_1 \oplus b_4$	$b_2 \oplus b_3 \oplus b_5$	$b_1 \oplus b_2 \oplus b_4$	$b_1 \oplus b_2 \oplus b_3 \oplus b_5$
6	01101	$b_1 \oplus b_4$	$b_2 \oplus b_3 \oplus b_5$	$b_1 \oplus b_2 \oplus b_4$	$b_1 \oplus b_2 \oplus b_3 \oplus b_5$	$b_1 \oplus b_4 \oplus b_5$
7	11010	$b_2 \oplus b_3 \oplus b_5$	$b_1 \oplus b_2 \oplus b_4$	$b_1 \oplus b_2 \oplus b_3 \oplus b_5$	$b_1 \oplus b_4 \oplus b_5$	$b_2 \oplus b_3 \oplus b_4 \oplus b_5$
8	10100	$b_1 \oplus b_2 \oplus b_4$	$b_1 \oplus b_2 \oplus b_3 \oplus b_5$	$b_1 \oplus b_4 \oplus b_5$	$b_2 \oplus b_3 \oplus b_4 \oplus b_5$	$b_1 \oplus b_2 \oplus b_3 \oplus b_4$
9	01001	$b_1 \oplus b_2 \oplus b_3 \oplus b_5$	$b_1 \oplus b_4 \oplus b_5$	$b_2 \oplus b_3 \oplus b_4 \oplus b_5$	$b_1 \oplus b_2 \oplus b_3 \oplus b_4$	$b_1 \oplus b_3 \oplus b_5$
10	10010	$b_1 \oplus b_4 \oplus b_5$	$b_2 \oplus b_3 \oplus b_4 \oplus b_5$	$b_1 \oplus b_2 \oplus b_3 \oplus b_4$	$b_1 \oplus b_3 \oplus b_5$	$b_4 \oplus b_5$
11	00100	$b_2 \oplus b_3 \oplus b_4 \oplus b_5$	$b_1 \oplus b_2 \oplus b_3 \oplus b_4$	$b_1 \oplus b_3 \oplus b_5$	$b_4 \oplus b_5$	$b_3 \oplus b_4$
12	01000	$b_1 \oplus b_2 \oplus b_3 \oplus b_4$	$b_1 \oplus b_3 \oplus b_5$	$b_4 \oplus b_5$	$b_3 \oplus b_4$	$b_2 \oplus b_3$
13	10000	$b_1 \oplus b_3 \oplus b_5$	$b_4 \oplus b_5$	$b_3 \oplus b_4$	$b_2 \oplus b_3$	$b_1 \oplus b_2$
14	00001	$b_4 \oplus b_5$	$b_3 \oplus b_4$	$b_2 \oplus b_3$	$b_1 \oplus b_2$	$b_1 \oplus b_2 \oplus b_5$
15	00010	$b_3 \oplus b_4$	$b_2 \oplus b_3$	$b_1 \oplus b_2$	$b_1 \oplus b_2 \oplus b_5$	$b_1 \oplus b_2 \oplus b_4 \oplus b_5$
16	00101	$b_2 \oplus b_3$	$b_1 \oplus b_2$	$b_1 \oplus b_2 \oplus b_5$	$b_1 \oplus b_2 \oplus b_4 \oplus b_5$	$b_1 \oplus b_2 \oplus b_3 \oplus \oplus b_4 \oplus b_5$
17	01010	$b_1 \oplus b_2$	$b_1 \oplus b_2 \oplus b_5$	$b_1 \oplus b_2 \oplus b_4 \oplus b_5$	$b_1 \oplus b_2 \oplus b_3 \oplus \oplus b_4 \oplus b_5$	$b_1 \oplus b_3 \oplus b_4 \oplus b_5$
18	10101	$b_1 \oplus b_2 \oplus b_5$	$b_1 \oplus b_2 \oplus b_4 \oplus b_5$	$b_1 \oplus b_2 \oplus b_3 \oplus \oplus b_4 \oplus b_5$	$b_1 \oplus b_3 \oplus b_4 \oplus b_5$	$b_3 \oplus b_4 \oplus b_5$
19	01011	$b_1 \oplus b_2 \oplus b_4 \oplus b_5$	$b_1 \oplus b_2 \oplus b_3 \oplus \oplus b_4 \oplus b_5$	$b_1 \oplus b_3 \oplus b_4 \oplus b_5$	$b_3 \oplus b_4 \oplus b_5$	$b_2 \oplus b_3 \oplus b_4$
20	10111	$b_1 \oplus b_2 \oplus b_3 \oplus \oplus b_4 \oplus b_5$	$b_1 \oplus b_3 \oplus b_4 \oplus b_5$	$b_3 \oplus b_4 \oplus b_5$	$b_2 \oplus b_3 \oplus b_4$	$b_1 \oplus b_2 \oplus b_3$
21	01110	$b_1 \oplus b_3 \oplus b_4 \oplus b_5$	$b_3 \oplus b_4 \oplus b_5$	$b_2 \oplus b_3 \oplus b_4$	$b_1 \oplus b_2 \oplus b_3$	$b_1 \oplus b_5$
22	11101	$b_3 \oplus b_4 \oplus b_5$	$b_2 \oplus b_3 \oplus b_4$	$b_1 \oplus b_2 \oplus b_3$	$b_1 \oplus b_5$	$b_2 \oplus b_4 \oplus b_5$
23	11011	$b_2 \oplus b_3 \oplus b_4$	$b_1 \oplus b_2 \oplus b_3$	$b_1 \oplus b_5$	$b_2 \oplus b_4 \oplus b_5$	$b_1 \oplus b_3 \oplus b_4$
24	10110	$b_1 \oplus b_2 \oplus b_3$	$b_1 \oplus b_5$	$b_2 \oplus b_4 \oplus b_5$	$b_1 \oplus b_3 \oplus b_4$	$b_3 \oplus b_5$
25	01100	$b_1 \oplus b_5$	$b_2 \oplus b_4 \oplus b_5$	$b_1 \oplus b_3 \oplus b_4$	$b_3 \oplus b_5$	$b_2 \oplus b_4$
26	11000	$b_2 \oplus b_4 \oplus b_5$	$b_1 \oplus b_3 \oplus b_4$	$b_3 \oplus b_5$	$b_2 \oplus b_4$	$b_1 \oplus b_3$
27	10001	$b_1 \oplus b_3 \oplus b_4$	$b_3 \oplus b_5$	$b_2 \oplus b_4$	$b_1 \oplus b_3$	b_5
28	00011	$b_3 \oplus b_5$	$b_2 \oplus b_4$	$b_1 \oplus b_3$	b_5	b_4
29	00111	$b_2 \oplus b_4$	$b_1 \oplus b_3$	b_5	b_4	b_3
30	01111	$b_1 \oplus b_3$	b_5	b_4	b_3	b_2

Положительным качеством формирователя кодов Галуа, который реализуется регистром сдвига на D-триггерах, является высокое быстродействие и независимость скорости генерирования битов Галуа от порядков примитивного полинома и разрядности регистра сдвига. Реализация арифметических операций в кодах поля Галуа выполняется на базе логических уравнений, формализация которых для 5-разрядного кода приведена в табл. 1.

Изложенный способ формализации логического описания кодонов поля Галуа предусматривает эмуляцию его работы исключительно программным путем, что не позволяет перейти к его аппаратной реализации. Особенно указанный недостаток проявляется при увеличение разрядности процессоров ВИТ от 32 до 64 бит. Данный способ алгебраического описания кодонов поля Галуа требует

использования больших объемов памяти для их хранения при выполнении операций сложения и умножения. Такое ограничение может быть в достаточной степени устранено путем использования рекуррентных свойств поля Галуа и генерирования элементов d_{ij} в процессе операции вертикального сложения и умножения. При этом применяются стартовые кодоны первого такта выполнения арифметических операций согласно выражению

$$b_i = d_{i,k} \cdot b_k \oplus d_{i,k-1} \cdot b_{k-1} \oplus \ldots \oplus d_{i,1} \cdot b_1, \qquad (3)$$

где $d_{i,k} \in \overline{0,1}$.

В работе [8] изложена теоретическая основа метода реализации арифметики суммирования чисел в базисе Галуа. Реализация операции сложения в поле Галуа сводится к выполнению над битами одного из слагаемых логических выражений операции по mod 2, которые представляют биты второго слагаемого. Это позволяет повысить быстродействие операции суммирования на один или два порядка по сравнению с суммированием в двоичной системе в результате исключения междуразрядных переносов. При этом вычисление каждого результата операции суммирования осуществляется за несколько тактов и не зависит от разрядности процессора ВИТ.

Рассмотрим операцию сложения двух чисел: $X_{(10)} = 13$, $Y_{(10)} = 14$ в поле Галуа $GF\begin{pmatrix}5\\2\end{pmatrix}$, алгоритм выполнения которой представим следующим образом:

1) первый операнд $X_{(10)} = 13$ представляется кодом Галуа $X_G = (10000)$ согласно табл. 1: $b_5 = 1$; $b_4 = 0$; $b_3 = 0$; $b_2 = 0$; $b_1 = 0$;

2) второй операнд $Y_{(10)} = 14$ представляется набором логических выражений $b_4 \oplus b_5$; $b_3 \oplus b_4$; $b_2 \oplus b_3$; $b_1 \oplus b_2$; $b_1 \oplus b_2 \oplus b_5$ (см. табл. 1);

3) выполняется операция сложения путем параллельного вычисления логических выражений

$$b_5 = b_4 \oplus b_5 = 1 \oplus 0 = 1;$$

$$b_4 = b_3 \oplus b_4 = 0 \oplus 0 = 0;$$

$$b_3 = b_2 \oplus b_3 = 0 \oplus 0 = 0;$$

$$b_2 = b_1 \oplus b_2 = 0 \oplus 0 = 0;$$

$$b_1 = b_1 \oplus b_2 \oplus b_5 = 0 \oplus 0 \oplus 1 = 1.$$

В результате получим

	b_5	b_4	b_3	b_2	b_1
$X =$	1	0	0	0	0
$Y =$	$b_4 \oplus b_5$	$b_3 \oplus b_4$	$b_2 \oplus b_3$	$b_1 \oplus b_2$	$b_1 \oplus b_2 \oplus b_5$
$X + Y =$	1	0	0	0	1.

Полученное значение 10001 в коде Галуа равно числу $27_{(10)}$ согласно табл. 1.

Важным преимуществом операции суммирования чисел в кодах Галуа по отношению к базису Радемахера является возможность повышения быстродействия операций инкремента и декремента. Такие операции выполняются, например, в адресных счетчиках при обращении к памяти. Сравнение выполнения операций инкремента в базисе Радемахера и Галуа свидетельствует о повышении быстродействия данной операции в базисе Галуа в n раз по отношению к базису Радемахера.

Базис Радемахера:

$$\begin{array}{r} \overset{\leftarrow\leftarrow\leftarrow \quad \leftarrow\leftarrow}{1\ 1\ 1 \ldots 1\ 1} \\ 0\ 0\ 0 \ldots 0\ 1 \\ \hline 1\ 0\ 0\ 0 \ldots 0\ 0 \end{array}$$

Базис Галуа:

$$\begin{array}{r} \boxed{1\boxed{11 \ldots 1\ 1}} \\ 111 \ldots 1\boxed{0}\leftarrow \end{array}$$

Здесь знак ← представляет символ сквозного переноса в следующий разряд.

Выполнение операции сложения для аппаратной реализации полного сумматора в базисе Галуа путем создания матрицы коэффициентов d_{ij} отражено в табл. 2.

Таблица 2. Матрица коэффициентов d_{ij}

Номер пп.	Код Галуа	Разряды кодов Галуа, выраженные через d_{ij}				
		d_{i5}	d_{i4}	d_{i3}	d_{i2}	d_{i1}
0	11111	10000	01000	00100	00010	00001
1	11110	01000	00100	00010	00001	10010
2	11100	00100	00010	00001	10010	01001
3	11001	00010	00001	10010	01001	10110
4	10011	00001	10010	01001	10110	01011
5	00110	10010	01001	10110	01011	10111
6	01101	01001	10110	01011	10111	11001
7	11010	10110	01011	10111	11001	11110
8	10100	01011	10111	11001	11110	01111
9	01001	10111	11001	11110	01111	10101
10	10010	11001	11110	01111	10101	11000
11	00100	11110	01111	10101	11000	01100
12	01000	01111	10101	11000	01100	00110
13	10000	10101	11000	01100	00110	00011
14	00001	11000	01100	00110	00011	10011
15	00010	01100	00110	00011	10011	11011
16	00101	00110	00011	10011	11011	11111
17	01010	00011	10011	11011	11111	11101
18	10101	10011	11011	11111	11101	11100
19	01011	11011	11111	11101	11100	01110
20	10111	11111	11101	11100	01110	00111
21	01110	11101	11100	01110	00111	10001
22	11101	11100	01110	00111	10001	11010
23	11011	01110	00111	10001	11010	01101
24	10110	00111	10001	11010	01101	10100
25	01100	10001	11010	01101	10100	01010
26	11000	11010	01101	10100	01010	00101
27	10001	01101	10100	01010	00101	10000
28	00011	10100	01010	00101	10000	01000
29	00111	01010	00101	10000	01000	00100
30	01111	00101	10000	01000	00100	00010

Согласно этой таблице и выражению (3) разработана базовая структурная компонента сумматора на основе суммирования двух операндов: $X_{(10)}$, $Y_{(10)}$. Рассмотрим выполнение операции суммирования двух чисел: $X_{(10)} = 8$; $Y_{(10)} = 9$. Значение X_G отвечает коду $b_5 = 0$; $b_4 = 1$; $b_3 = 0$; $b_2 = 1$; $b_1 = 1$, а код Y_G согласно

табл. 2 представлен логическими выражениями X_G: $b_1 \oplus b_2 \oplus b_3 \oplus b_5$; $b_1 \oplus b_4 \oplus b_5$; $b_2 \oplus b_3 \oplus b_4 \oplus b_5$; $b_1 \oplus b_2 \oplus b_3 \oplus b_4$; $b_1 \oplus b_3 \oplus b_5$, что соответствует кодам d_{ij}: 10111; 11001; 11110; 01111; 10101. Результат сложения данных чисел получен логической обработкой кодов X_G и коэффициентов d_{ij}, которые отвечают коду Y_G:

$$G_5 = 1 \wedge b_5 \oplus 0 \wedge b_4 \oplus 1 \wedge b_3 \oplus 1 \wedge b_2 \oplus 1 \wedge b_1 = 1 \wedge 1 \oplus 0 \wedge 0 \oplus 1 \wedge 1 \oplus 1 \wedge 0 \oplus 1 \wedge 0 = 0;$$

$$G_4 = 1 \wedge b_5 \oplus 1 \wedge b_4 \oplus 0 \wedge b_3 \oplus 0 \wedge b_2 \oplus 1 \wedge b_1 = 1 \wedge 1 \oplus 1 \wedge 0 \oplus 0 \wedge 1 \oplus 0 \wedge 0 \oplus 1 \wedge 0 = 1;$$

$$G_3 = 1 \wedge b_5 \oplus 1 \wedge b_4 \oplus 1 \wedge b_3 \oplus 1 \wedge b_2 \oplus 0 \wedge b_1 = 1 \wedge 1 \oplus 1 \wedge 0 \oplus 1 \wedge 1 \oplus 1 \wedge 0 \oplus 0 \wedge 0 = 0;$$

$$G_2 = 0 \wedge b_5 \oplus 1 \wedge b_4 \oplus 1 \wedge b_3 \oplus 1 \wedge b_2 \oplus 1 \wedge b_1 = 0 \wedge 1 \oplus 1 \wedge 0 \oplus 1 \wedge 1 \oplus 1 \wedge 0 \oplus 1 \wedge 0 = 1;$$

$$G_1 = 1 \wedge b_5 \oplus 0 \wedge b_4 \oplus 1 \wedge b_3 \oplus 0 \wedge b_2 \oplus 1 \wedge b_1 = 1 \wedge 1 \oplus 0 \wedge 0 \oplus 1 \wedge 1 \oplus 0 \wedge 0 \oplus 1 \wedge 0 = 0.$$

На рис. 2 показана логическая схема полного многоразрядного сумматора, согласно которой реализуется операция сложения кодов Галуа.

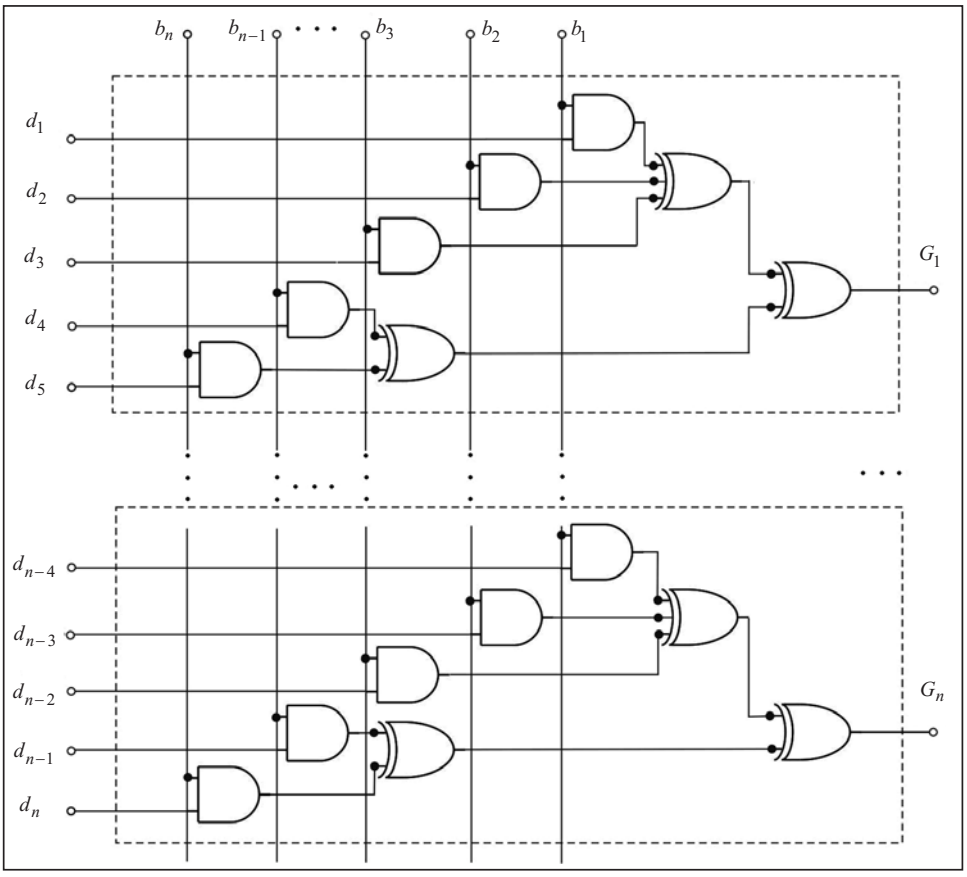


Рис. 2. Базовая структурная компонента сумматора Галуа

Операция умножения чисел $X \cdot Y$ в кодах поля Галуа может быть выполнена двумя способами:

1) X-кратным сложением числа X или Y и суммы этих чисел;

2) последовательным сложением удвоенных значений числа X согласно двоичному представлению числа Y в базисе Радемахера.

Принцип умножения для поля Галуа $GF\binom{5}{2}$: $X = 4_{(10)} = 10011$, $Y = 3_{(10)} =$ $= b_i \oplus$.

Вначале выполняем операцию сложения $X + X$ для поля Галуа $GF\binom{5}{2}$, $X = 4_{(10)}$; $X = 4_{(10)}$:

	b_5	b_4	b_3	b_2	b_1
$X =$	1	0	0	1	1
$X =$	b_1	$b_2 \oplus b_5$	$b_1 \oplus b_4$	$b_2 \oplus b_3 \oplus b_5$	$b_1 \oplus b_2 \oplus b_4$
$X + X = Z$	1	0	1	0	0.

Полученное значение, равное $Z = 8_{(10)}$, суммируется с числом $X = 4_{(10)}$ следующим образом:

	b_5	b_4	b_3	b_2	b_1
$X =$	1	0	0	1	1
$Z =$	$b_1 \oplus b_2 \oplus b_4$	$b_1 \oplus b_2 \oplus b_3 \oplus b_5$	$b_1 \oplus b_4 \oplus b_5$	$b_2 \oplus b_3 \oplus b_4 \oplus b_5$	$b_1 \oplus b_2 \oplus b_3 \oplus b_4$
$X + Z = T$	0	1	0	0	0.

Значение $T = 01000$ в коде Галуа есть результат операции умножения $12_{(10)}$. В общем случае алгоритм выполнения операции умножения представляем следующим образом:

$$
\begin{aligned}
& x = (d0_n, d0_{n-1}, \ldots, d0_1) + (b_n \oplus, b_{n-1} \oplus, \ldots, b_1 \oplus) * Y_0, \\
& 2^1 x = (d1_n, d1_{n-1}, \ldots, d1_1) + (b_n \oplus, b_{n-1} \oplus, \ldots, b_1 \oplus) * Y_2, \\
& 2^2 x = (d2_n, d2_{n-1}, \ldots, d2_1), \\
& \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
& 2^{n-1} x = (dn_n, dn_{n-1}, \ldots, dn_1) + (b_n \oplus, b_{n-1} \oplus, \ldots, b_1 \oplus) * Y_n, \\
& X * Y = (dn_n, dn_{n-1}, \ldots, dn_1); \\
& X_{(G)} = d_{ij} = (d_n, d_{n-1}, \ldots, d_1); \\
& Y_{(G)} = Y_{n-1} * 2^{n-1} + Y_{n-2} * 2^{n-2} + \ldots + Y_1 * 2^1 + Y_0 * 2^0.
\end{aligned}
$$

Здесь $X_{(G)} = d_{ij} = (d_n, d_{n-1}, \ldots, d_1)$ — бит-ориентированный код числа X; $Y_{(G)} = Y_{n-1} * 2^{n-1} + Y_{n-2} * 2^{n-2} + \ldots + Y_1 * 2^1 + Y_0 * 2^0$ — двоичный код числа Y; $(b_n \oplus, b_{n-1} \oplus, \ldots, b_1 \oplus)$ — логические уравнения удвоенных значений кодов d_{ij}, которые выбираются из постоянного запоминающего устройства (ПЗУ).

Разработанный алгоритм выполнения операций сложения и умножения в базисе Галуа может быть оптимизирован путем представления значений кодов в системе исчисления остаточных классов ТЧБ Галуа–Крестенсона, что позволит реализовать многоразрядные процессоры в диапазоне чисел произведения системы взаимно простых модулей $P_1, P_2, \ldots, P_j, \ldots, P_k$ при разрядности кодов Галуа, не превышающей $\log_2 P_{j\max}$.

Разработанные теоретические положения и примеры реализации операций сложения и умножения создают возможность реализации структуры процессоров ВИТ базиса Галуа. На рис. 3 представлена структура взаимодействия компонентов процессора ВИТ, которая реализирует бит-ориентированное выполнение операций сложения и умножения. Здесь A — аккумулятор, М — мультиплексор, ПЗУ — постоянное запоминающее устройство, ОЗУ — оперативное запоминающее устройство, РОН — регистры общего назначения.

Пошаговое выполнение операции сложения над кодами Галуа с занесением результата с аккумулятора в ОЗУ выполняется следующим образом:

y_1) $G(B) := x, G(C) := y$ — занесение операндов x и y в регистры B и C ($G(B)$, $G(C)$ — бит-ориентированные коды поля Галуа);

y_2) $A := G(B)$ — присвоение аккумулятору значение регистра B;

y_3) $A := A + G(C)$ — добавление к аккумулятору значение регистра C;

y_4) занесение результата выполнения операции сложения в ОЗУ;

y_5) очистка содержимого аккумулятора.

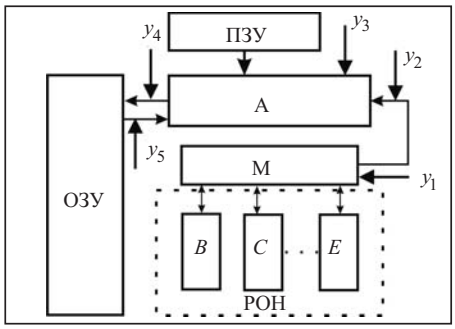


Рис. 3. Схема реализации операции сложения в базисе Галуа

ФУНКЦИОНАЛЬНАЯ СТРУКТУРА СПЕЦПРОЦЕССОРА НА ОСНОВЕ ВИТ

Вертикально-информационная технология, основой которой есть ТЧБ Галуа, направлена на совершенствование функциональных ограничений двоичной системы и развитие архитектур процессоров в арифметике бит-ориентированных кодов Галуа. В общем случае базовая структура чипа такого спецпроцессора может иметь не больше восьми внешних соединений, которые включают бит-ориентированные шины Y — управления; Ад — адреса; D — данных; c/s — выбора кристалла; W/R — ввода/вывода (рис. 4).

На основе структурной схемы внешних информационных связей процессора ВИТ, структура которого представлена на рис. 4, разработана функциональная схема спецпроцессора ВИТ, отраженная на рис. 5. Здесь АЛУ — арифметико-логическое устройство; ОЗУ — оперативное запоминающее устройство; РОН — регистры общего назначения; УУС — устройство управления и синхронизации; G_j — бит-ориентированный адрес в базисе Галуа; БВИ — буфер внешнего интерфейса; G_k — счетчик команд в базисе Галуа; D_i, D_j — бит-ориентированные данные; $G_0 \div G_n$ — адресный код поля Галуа; D_z, G_z, Y_z — внешние бит-ориентированные интерфейсные шины данных, адреса и управления.

Отличительной особенностью таких процессоров является адресация данных в ОЗУ счетчиком команд путем организации бит-ориентированных адресных данных однобитовыми инкрементными кодами Галуа, что дает преимущество по сравнению с базисом Радемахера, который требует формирования параллельных двоичных n-разрядных кодов. При этом, как показано в работе [9], существенно упрощается проектирование адресного дешифратора ОЗУ, а также возможность достаточно простой реализации памяти параллельного коллективного доступа, что особенно важно для мультипроцессорных и кластерных вычислительных систем.

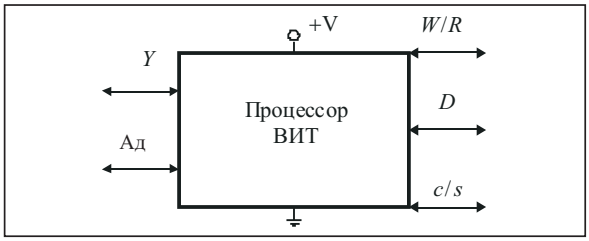


Рис. 4. Структура внешних информационных связей процессора ВИТ

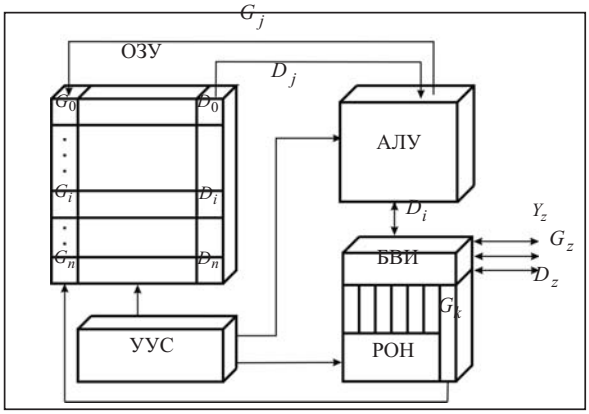


Рис. 5. Функциональная структура спецпроцессора ВИТ

ЗАКЛЮЧЕНИЕ

На основе рассмотренных методов сложения и умножения кодовых последовательностей Галуа разработана базовая логическая схема сумматора, в которой отсутствуют сквозные переносы между разрядами. Разработанная функциональная структура спецпроцессора ВИТ характеризуется минимальным количеством внешних информационных связей, бит-ориентированными шинами адреса, данных и управления, что позволит оптимизировать характеристики процессоров ВИТ и уменьшить габариты их кристаллов. Разработка и реализация процессоров ВИТ позволит расширить сферу применения спецпроцессоров для обработки бит-ориентированных потоков информации, а также использовать их в качестве компонентов мультибазисных RCG-процессоров.

СПИСОК ЛИТЕРАТУРЫ

1. Chiou-Yng Lee, Pramod Kumar Meher, Che Wun Chiou, Jim-Min Lin. Concurrent detection/correction in finite field architectures over GF(2m) // Cryptography Research Perspectives. — New York (USA): Nova Sci. Publ., 2008. — P. 49–96.
2. Николайчук Я.Н. Коды поля Галуа: Теория и применение. — Тернополь: ТЗОО «Терно граф», 2012. — 576 с.
3. Николайчук Я.Н. Теория источников информации. — Тернополь: ТЗОО «Терно граф», 2010. — 534 с.
4. Patent N6,523,054 B1 (USA). Galois field arithmetic processor / Shunsuke Kamijo — Fujitsu Limited, Kawasaki (Japan). — 2003. — Appl. N 09/437,473.
5. Patent N4,473,887 (USA). Processing circuit for operating on elements of a Galois field / Kentaro Odaka. — Sony Co., Tokyo (Japan). — 1984. — Appl. N 360,205.
6. Patent N7,082,452 B2 (USA) Galois field multiply/multiply-add/multiply accumulate / Yosi Stein, Haim Primo, Yaniv Sapir. — Analog Devices. — 2006. — Appl. N 10/228,526.
7. Patent N4,918,638 (USA). Multiplier in a Galois field / Michito Matsumoto, Kazuhiro Murase. — Matsushita Electric Industrial. — Osaka (Japan). — 1990. — Appl. N 107,363.
8. Николайчук Я.Н., Заставный О.М., Гуменный П.В. Теоретические основы и принципы построения арифметико-логического устройства на основе вертикально-информационной технологии // Вест. Хмельниц. нац. ун-та. — 2012. — № 2. — С. 190–196.
9. Гуменный П.В. Структура и системные характеристики многопортовой ПКД на основе вертикально информационной технологии в базисе Галуа // Сб. науч. тр. Бучацкого ин-та менеджмента и аудита. — 2010. — **1**. — Вып. 6. — С. 71–75.
10. Глухов В.С., Ногаль М.В. Специализированный одноразрядный процессор для защиты информации в гарантировано устойчивых системах // Радиоэлектрон. и компьютер. системы. — Харьков: ХАИ, 2008. — С. 104–109.
11. Drozd A., Antoshchuk S. New on-line testing methods for approximate data processing in the computing circuits // 6th IEEE Intern. Conf. on Intellig. Data Acquisition and Adv. Comput. Syst.: Technology and Applications (Prague, Czech. Republic), 15–17 Sept., 2011. — P. 291–294.
12. Сергиенко И.В., Гупал А.М., Вагис А.А. Соотношения комплементарности в записи оснований по одной нити в хромосомах ДНК // Проблемы управления и информатики. — 2005. — № 4. — С. 153–157.
13. Chiou C.W., Lee Chiou-Yng, Deng An-Wen, Lin Jim-Min. Concurrent error detection in Montgomery multiplication over GF(2^m) // IEICE Trans. on Fundamentals. — 2006. — **E89-A**, N 2. — P. 566–574.
14. Fan H. New GF(2^n) parallel multiplier using redundant representation. Researches in GF(2^n) // Multiplication Algorithms, PHD. — Tsinghua Univ. (Chinese). — http://eprint.iarc.org/2004/137.
15. Lee J.-W., Meher C.-Y., P.K. Patra J.C. Concurrent error detection in bit-serial normal basis multiplication over GF(2^m). Using Multiple Parity Prediction Schemes in Large Scale Integration (VLSI) Systems // IEEE Transac. on Very Large Scale Integration (VLSI) Systems, 25 August, 2009.
16. А.с. N 70744 U Украина H038M. Аналого-цифровой преобразователь / Я.Н. Николайчук, П.В. Гуменный // Бюл. № 12. — 2012.
17. Круцкевич Н.Д., Николайчук Я.Н. Принципы построения RCG процессора // Тез. междунар. науч.-техн. конф. «Контроль и управление в сложных системах» (КУСС-2003). — Винница: УНИВЕРСУМ-Винница, 2003. — С. 73

Поступила 10.04.2013