«ФИЗИКА ЭЛЕМЕНТАРНЫХ ЧАСТИЦ И АТОМНОГО ЯДРА»
1993, ТОМ 24, ВЫП.1

УДК 681.325.67: 621.374.32

МЕТОД СИНДРОМНОГО КОДИРОВАНИЯ И ЕГО ПРИМЕНЕНИЕ ДЛЯ БЫСТРОГО ОТБОРА СОБЫТИЙ В ЭКСПЕРИМЕНТАХ ПО ФИЗИКЕ ВЫСОКИХ ЭНЕРГИЙ

Н.М.Никитюк

Объединенный институт ядерных исследований, Дубна

Рассмотрены вопросы применения теории и практики алгебраической теории кодирования для сжатия и обработки данных, регистрируемых в многоканальных детекторах заряженных частиц. Описана система аналогий, как оказалось, существующая между теорией корректирующих кодов и теорией годоскопических систем. Основным результатом такого подхода явилось создание принципиально новых логических устройств, таких как параллельные шифраторы на $t > 1$ сигналов, мажоритарные схемы совпадений на большое число входов, динамически программируемые модули и специализированные процессоры с алгебраической структурой, оперирующие над элементами поля Галуа $GF(2^m)$, для быстрого отбора физических событий. Показано также, каким образом, используя вычисления в полях Галуа, можно создавать динамически программно-управляемые логические модули, которые имеют перспективы применения в триггерных системах.

The problems are described of using the theory and the practice of the algebraic coding theory for data compression and processing in multichannel charge particle detectors. A system of analogies between the theories of correcting codes and hodoscopic systems is considered. The main result of this approach is the development of radically new electronic logic units, i.e., parallel encoders for $t > 1$ events, majority coincidence circuits, dynamically programmable logic modules and special-purpose processors with algebraic structure (in the Galois field $GF(2^m)$) for a fast event selection in nuclear physics experiments.

ВВЕДЕНИЕ

Несмотря на простоту математической структуры, поля Галуа имеют широкие перспективы применения как в информатике, так и в других областях науки и техники, например, в электронных методах в физике высоких энергий. На рис.1 показаны области применения алгебры конечных полей. Более 30 лет назад У.Питерсон [1] применил теорию поля Галуа для декодирования кодов, исправляющих $t > 1$ ошибок. Это направ-

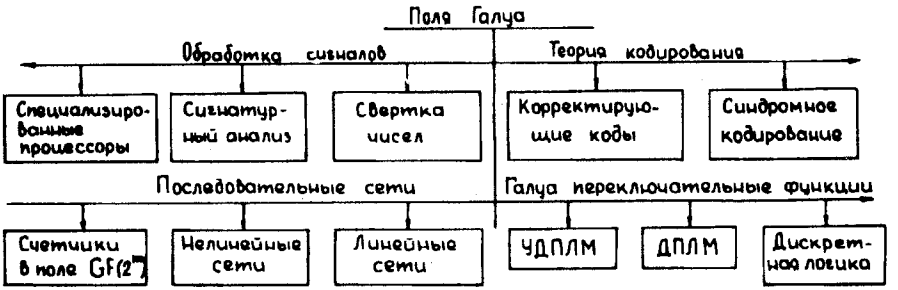


Рис.1. Применение алгебры Галуа в различных областях науки и техники. УДПЛМ — универсальный динамически программируемый логический модуль. ДПЛМ — динамически программируемый логический модуль

ление успешно используется до нашего времени. В свою очередь, алгебраическая теория кодирования стимулировала создание трех новых направлений в современной информатике: кодирование источников [2], сигнатурный анализ [3], применяемый для создания систем тестирования микропроцессоров и больших интегральных микросхем, и метод синдромного кодирования [4], предложенный автором для быстрой компрессии физических данных и для создания не имеющих аналогов в мировой практике параллельных шифраторов для быстрого кодирования $t > 1$ сигналов с помощью комбинационных схем [5,6]. Ряд теорем и идей алгебраической теории кодирования автор применил для создания мажоритарных схем совпадений на большое число входов ($n > 100$), специализированных процессоров для быстрого отбора событий в спектрометрах, применяемых в физике высоких энергий, преобразователей время — код и параллельных шифраторов для световой кодировки устройств сжатия и обработки данных, регистрируемых в двухкоординатных детекторах [7—9]. Данный метод использовался также для построения триггерного процессора [10].

Известны также и другие области применения полей Галуа: преобразование Фурье [11], последовательностные автоматы, где входы и выходы кодируются элементами поля Галуа, играющими важную роль в вычислительной технике [12—16] и ядерной электронике [9].

Цель данного обзора состоит в систематизации работ по развитию и применению метода синдромного кодирования для компрессии и обработки данных, регистрируемых в многоканальных детекторах заряженных частиц. В основу метода положена алгебраическая теория кодирования, которая, в свою очередь, базируется на методах алгебры полей Галуа.

ОПЕРАЦИИ НАД ЭЛЕМЕНТАМИ ПОЛЯ ГАЛУА

В отличие от обычной и широко известной двоичной арифметики, в методе синдромного кодирования арифметические и алгебраические операции выполняются над расширенным полем Галуа $GF(p^m)$, где p — простое число, называемое характеристикой поля. Для двоичной системы $p = 2$ и m — целое число. Правилам выполнения операций над элементами поля посвящен ряд работ [17—19]. С целью облегчения чтения данного обзора для читателей, не знакомых с методами выполнения арифметических и алгебраических операций над элементами поля Галуа $GF(2^m)$, ниже приводится необходимая систематизированная информация.

Элементы поля Галуа можно складывать, умножать, инвертировать и т.д. так же, как и обычные, более известные рациональные, действительные и комплексные числа. Принципиальная разница заключается в том, что эти числа содержат бесконечное число элементов, в то время как поле Галуа, по определению, содержит только конечное число элементов [13] $n = p^m$. Поле, содержащее p^m элементов обозначается $GF(2^m)$. Как следствие этого, поле Галуа $GF(2^1)$ содержит два элемента: 0 и 1. Количество ненулевых элементов конечного поля равно степени его характеристики, т.е. $n = 2^m - 1$, причем число различных элементов поля называется его порядком. Все элементы данного поля могут быть просто получены с помощью неприводимого полинома. Таблицы таких полиномов вплоть до 34-й степени приведены в [1, прил. В]. В дальнейшем для построения различных устройств компрессии и обработки данных, регистрируемых в многоканальных детекторах заряженных частиц, содержащих $n = 15$, 31 и 63 каналов, мы будем использовать неприводимые полиномы 4, 5 и 6-й степени: $X^4 + X + 1$, $X^5 + X^2 + X + 1$ и $X^6 + X + 1$. Следует отметить, что знак + обозначает в наших примерах суммирование по модулю два.

Среди элементов поля имеется m линейно независимых (базисных) элементов. Например, для полинома $f(X) = X^6 + X + 1$ ($m = 6$) имеем следующие базисные элементы: $a^0 = 100000$; $a^1 = 010000$; $a^2 = 001000$; $a^3 = 000100$; $a^4 = 000010$ и $a^5 = 000001$. Один из них — элемент a^1 — является корнем полинома $f(X)$. Следовательно, каждый ненулевой элемент поля может быть представлен как степень элемента a^1. Другими словами, мультипликативная группа конечного поля носит циклический характер. Далее, наименьшее положительное число, для которого

$a^n = a^0 = 1$, называется порядком элемента a^1. Если же порядок элемента a^1 равен n, то элементы $a^0, a^1, a^2, \ldots a^{n-1}$ различны. Так, в нашем примере $n = 2^m - 1 = 63$. Следовательно, например, $a^{126} = (a^{63})^2 = a^0$, $a^{64} = a^{63}$, $a^1 = a^1$ и т.д. Учитывая, что a^1 суть корень полинома $f(X)$, остальные элементы поля $GF(2^6)$ можно получить из соотношения $a^6 + a^1 + 1 = 0$, т.е. $a^6 = a^1 + 1 = 110000$, $a^7 = a^2 + a^1 = 011000$ и т.д. Перечень всех элементов данного поля приведен в приложении.

Нередко какой-либо элемент поля A удобно представлять в виде полинома степени $m - 1$ (вектора): $A = d_0a^0 + d_1a^1 + d_2a^2 + \ldots + d_{m-2}a^{m-2} + d_{m-1}a^{m-1}$, где коэффициенты d_0, d_1, d_2 ... d_{m-2}, d_{m-1} равны 0 или 1. Так, для элемента $a^9 = a^4 + a^3$ $d_0 = d_1 = d_2 = d_5$ и $d_3 = d_4 = 1$. Так же, как и в обычной арифметике, в поле Галуа имеются отличия выполнения операций над элементами поля машинным способом и вручную. Это прежде всего касается таких операций, как умножение, деление, возведение в степень и извлечение квадратного корня. Операция умножения вручную выполняется просто: степень произведения равна сумме степеней сомножителей (с учетом цикличности поля). Операция деления элемента A на элемент B эквивалентна умножению элемента A на инверсный элемент B^{-1}. В свою очередь, инверсный элемент B^{-1} элемента B находится из соотношения: $BB^{-1} = 1 = a^0$. Например, для элемента a^9 элемент a^{54} является инверсным элементом, так как $a^9a^{54} = a^{63} = a^0$. При вычислении вручную можно пользоваться простым правилом: степень инверсного элемента, имеющего степень k, равна $2^m - 1 - k$. Например, степень инверсного элемента a^{25} равна 38. Далее, рассмотрим извлечение корня. Если i — число четное, то $(a^i)^{1/2} = a^{i/2}$. Если же i — число нечетное, то степень элемента, соответствующего $(a^i)^{1/2}$, равна $(i + 2^m - 1)/2$. Например, $(a^5)^{1/2} = a^{34}$. Операция возведения в степень элементов поля выполняется так же, как и возведение в степень обычных чисел, с той лишь разницей, что она выполняется по модулю n. Например, $(a^{60})^{12} = a^{720} = a^{63\times11}$ $a^{27} = a^{27}$. Такие операции, как сложение и вычитание в поле $GF(2^m)$, равносильны и выполняются по модулю два.

АППАРАТНАЯ РЕАЛИЗАЦИЯ НЕКОТОРЫХ ОПЕРАЦИЙ В ПОЛЕ ГАЛУА $GF(2^m)$

Счетчики. Для получения последовательности элементов поля как в прямом, так и в обратном направлении применяются сдвиговые регистры с логической обратной связью [1]. Структура связей в таких счетчиках зависит от неприводимого полинома, выбранного для построения поля. На рис.2 приведены счетчики в поле Галуа для счета в прямом и обратном направлении по модулю многочлена $X^6 + X + 1$ соответственно. Если в младший разряд регистра, представленного на рис.2,*а*, поместить единицу, а в остальные нули, то последовательные сдвиги регистра дадут представление последовательных степеней элемента a^1 в форме, в какой они приведены в приложении. Наличие обратной связи из старшего разряда в младшие позволяет получить значение $a^6 = a^1 + 1$. Сдвиг влево (рис.2,*б*) соответствует делению на a^1 так, что единица переноса, выходящая из ячейки младшего разряда, дает значение $a = 1 + a^5$ [6].

Умножение и возведение в степень. Умножение двух элементов поля при заданном базисе можно выполнить, если эти элементы представить в виде полиномов [19]. Так, если один элемент поля A представить в виде

$$A = a^0 a_0 + a^1 a_1 + a^2 a_2 + a^3 a_3 + a^4 a_4 + a^5 a_5,$$

а другой элемент в виде

$$B = a^0 b_0 + a^1 b_1 + a^2 b_2 + a^3 b_3 + a^4 b_4 + a^5 b_5,$$

то прямое умножение этих полиномов по модулю 6 даст произведение двух элементов поля $GF(2^6)$, которое просто реализуется при помощи логических элементов И и схем проверки на четность. Эти схемы, по существу, представляют собой многовходовые сумматоры по модулю два.

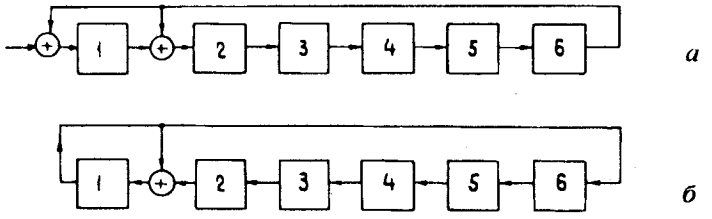


Рис.2. Счетчики в поле Галуа по модулю полинома $X^6 + X + 1$: *а* — счетчик в прямом направлении; *б* — счетчик в обратном направлении; 1÷6 — ячейки сдвигового регистра, ⊕ — сумматоры по модулю два

Обозначая соответствующие координаты элемента произведения через g_0, g_1, g_2, g_3, g_4 и g_5, получим следующие булевы выражения для произведения двух элементов:

$$g_0 = a_0b_0 + a_1b_5 + a_2b_4 + a_3b_3 + a_4b_2 + a_5b_1,$$

$$g_1 = a_2b_4 + a_2b_5 + a_3b_3 + a_3b_4 + a_4b_2 + a_5b_2 + a_0b_1 + a_1b_0 + a_1b_5 + a_4b_3 + a_5b_1,$$

$$g_2 = a_0b_2 + a_1b_1 + a_3b_4 + a_3b_5 + a_4b_3 + a_4b_4 + a_5b_2 + a_5b_3 + a_2b_0 + a_2b_5,$$

$$g_3 = a_0b_3 + a_1b_2 + a_3b_0 + a_3b_5 + a_4b_4 + a_4b_5 + a_5b_4 + a_5b_3 + a_2b_1, \quad (1)$$

$$g_4 = a_0b_4 + a_1b_3 + a_2b_2 + a_3b_1 + a_4b_0 + a_4b_5 + a_5b_4 + a_5b_5,$$

$$g_5 = a_0b_5 + a_1b_4 + a_2b_3 + a_3b_2 + a_4b_1 + a_5b_0 + a_5b_5.$$

Если в равенствах (1) положить $A = B$, то получим булевы выражения для возведения элемента поля (допустим это элемент B) в квадрат:

$$g_0 = b_0 + b_3,\ g_1 = b_3,\ g_2 = b_1 + b_4,\ g_3 = b_4,\ g_4 = b_2 + b_5 \text{ и } g_5 = b_5. \quad (2)$$

Путем итерации с помощью ЭВМ нетрудно получить булевы выражения для возведения любого элемента поля в более высокие степени.

Извлечение квадратного корня. Извлечение квадратного корня и решение уравнения второй степени в поле характеристики 2 являются линейными операциями и поэтому аппаратно реализуются довольно просто. Например, имеют место соотношения $(X + Y)^2 = X^2 + Y^2$ и $(X + Y)^{1/2} = X^{1/2} + Y^{1/2}$ [20]. Для извлечения квадратного корня из элемента A необходимо от одного базиса $a^0, a^1, a^2, a^3, a^4, a^5$ перейти к другому базису $(a^0)^{1/2} = a^0$; $(a^1)^{1/2} = a^{32}$; $(a^2)^{1/2} = a^1$; $(a^3)^{1/2} = a^{33}$; $(a^4)^{1/2} = a^2$, $(a^5)^{1/2} = a^{34}$. Тогда выражение для извлечения квадратного корня из элемента B получается из матричного уравнения

$$\left|b_0, b_1, b_2, b_3, b_4, b_5\right| \times \begin{vmatrix} a^0 \\ a^{32} \\ a^1 \\ a^{33} \\ a^2 \\ a^{34} \end{vmatrix} = A^{1/2}.$$

Например, $(a^{25})^{1/2} = a^{44}$ или

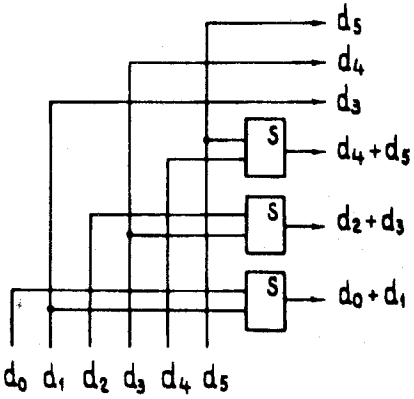


Рис.3. Принципиальная схема для извлечения второй степени из произвольного элемента в поле $GF(2^6)$. S — микросхема SN7486

$$[010001] \times \begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{vmatrix} = a^{44}.$$

На рис.3 в качестве примера приведена принципиальная схема вычисления квадратного корня из любого элемента поля $GF(2^6)$. Методы решения уравнений в поле Галуа будут рассмотрены ниже.

Вычисление инверсного элемента. Для вычисления инверсного элемента B^{-1} элемента B необходимо последний возвести в степень $2^m - 2$, так как $BB^{2m-2} = B^{2^m-1} = 1$. Еще проще инверсный элемент можно вычислить с помощью программируемой памяти, т.е. методами табличной арифметики.

ПЕРЕКЛЮЧАТЕЛЬНЫЕ ФУНКЦИИ ГАЛУА

Переключательная функция Галуа над полем $GF(p^n)$ (что весьма важно) может быть представлена в виде полинома степени меньшей, чем p^n [12,13]. Такое представление широко применяется в теории переключательных функций Галуа (ГПФ). Поскольку поле Галуа $GF(2^m)$ является естественным расширением булева поля, то представление переключательных функций в виде полинома, где как переменные, так и коэффициенты суть элементы поля, имеет ряд преимуществ принципиального характера.

1. Над ГПФ можно выполнять алгебраические операции, что упрощает проблему минимизации и ее формального представления.

2. В поле Галуа имеются такие операции, как сложение, умножение и деление, которые дают дополнительные преимущества перед булевым полем [16].

3. Представление переключательных функций в виде полиномов позволяет использовать стандартные системы программирования для расчета сложных логических устройств.

Таблица 1. Элементы поля $GF(2^3)$ $(a^2 + a + 1)$

Элементы	Разложение по базисным элементам			Двоичные эквиваленты		Минтермы
0	0			000	$GF(2)$	$\overline{X}_0\overline{X}_1\overline{X}_2$
a^0	1			100		$X_0\overline{X}_1\overline{X}_2$
a^1		a^1		010		$\overline{X}_0X_1\overline{X}_2$
a^2			a^2	001		$\overline{X}_0\overline{X}_1X_2$
a^3	1	$+a^1$		110	$GF(2^3)$	$X_0X_1\overline{X}_2$
a^4		a^1	$+a^2$	011		$\overline{X}_0X_1X_2$
a^5	1	$+a^1$	$+a^2$	111		$X_0X_1X_2$
a^6	1		$+a^2$	101		$X_0\overline{X}_1X_2$
$a^7 = a^0$	1			100		$X_0\overline{X}_1\overline{X}_2$

4. Поскольку состояния входов и выходов логического устройства кодируются элементами поля, то следующее состояние можно представить как полиномиальную функцию текущего состояния и текущего выхода.

5. Описание многозначных и многоуровневых схем имеет весьма компактный вид, что облегчает их теоретическое исследование.

Ниже будут изложены новые результаты, полученные автором, по практическому применению ГПФ, с целью создания быстрых программируемых модулей и процессорных устройств для электронных методов в физике высоких энергий. Кроме того, будут рассмотрены также разработанные быстрые алгоритмы и устройства для выполнения комбинированных операций над элементами поля Галуа $GF(2^m)$. С этой целью были использованы аналитические выкладки и вычисления на ЭВМ с применением таких систем программирования, как PL/1, REDUCE и SCOONSCHIP [21—23]. Читателям, интересующимся более детально свойствами ГПФ, можно рекомендовать литературу [24—28].

В табл.1 и 2 в качестве примера даны три возможных способа представления элементов поля $GF(2^3)$ и $GF(2^4)$.

Рассмотрим несколько примеров применения ГПФ. Известно, что любую ГПФ $F(X) = F(X_0, X_1, X_2, \ldots, X_{m-1})$ m аргументов в поле $GF(2^m)$ можно представить в виде полинома [13]:

$$F(X) = B(0) + A(1)X + A(2)X^2 + A(3)X^3 + \ldots + A(2^m - 1)^{2^{m-1}}. \quad (3)$$

Таблица 2. Элементы поля $GF(2^4)$ $(a^4 + a^1 + 1)$

Элементы	Разложение по базисным элементам	Двоичные эквиваленты		Минтермы
0	0	0000	$GF(2)$	$\bar{X}_0\bar{X}_1\bar{X}_2\bar{X}_3$
a^0	1	1000	$GF(2)$	$X_0\bar{X}_1\bar{X}_2\bar{X}_3$
a^1	a^1	0100	$GF(2^4)$	$\bar{X}_0X_1\bar{X}_2\bar{X}_3$
a^2	a^2	0010	$GF(2^4)$	$\bar{X}_0\bar{X}_1X_2\bar{X}_3$
a^3	a^3	0001	$GF(2^4)$	$\bar{X}_0\bar{X}_1\bar{X}_2X_3$
a^4	$1 + a^1$	1100	$GF(2^4)$	$X_0X_1\bar{X}_2\bar{X}_3$
a^5	$a^1 + a^2$	0110	$GF(2^4)$	$\bar{X}_0X_1X_2\bar{X}_3$
a^6	$a^2 + a^3$	0011	$GF(2^4)$	$\bar{X}_0\bar{X}_1X_2X_3$
a^7	$1 + a^1 + a^3$	1101	$GF(2^4)$	$X_0X_1\bar{X}_2X_3$
a^8	$1 + a^2$	1010	$GF(2^4)$	$X_0\bar{X}_1X_2\bar{X}_3$
a^9	$a^1 + a^3$	0101	$GF(2^4)$	$\bar{X}_0X_1\bar{X}_2X_3$
a^{10}	$1 + a^1 + a^2$	1110	$GF(2^4)$	$X_0X_1X_2\bar{X}_3$
a^{11}	$a^1 + a^2 + a^3$	0111	$GF(2^4)$	$\bar{X}_0X_1X_2X_3$
a^{12}	$1 + a^1 + a^2 + a^3$	1111	$GF(2^4)$	$X_0X_1X_2X_3$
a^{13}	$1 + a^2 + a^3$	1011	$GF(2^4)$	$X_0\bar{X}_1X_2X_3$
a^{14}	$1 + a^3$	1001	$GF(2^4)$	$X_0\bar{X}_1\bar{X}_2X_3$
a^{15}	1	1000	$GF(2^4)$	$X_0\bar{X}_1\bar{X}_2\bar{X}_3$

Коэффициенты $A(k)$ вычисляются из равенства

$$A(k) = \sum_{i=1}^{2^m-1} a_i^{-k}[\,B(0) + B(a_j)\,]\,, \quad k = 1, 2, 3, \ldots, 2^{m-1},$$

где $B(a_j)$ — элементы подстановки, получаемые из таблицы входов и выходов, и $B(0)$ — значение функции в нулевой точке. Следовательно, можно предложить следующую последовательность синтеза ГПФ переменных:

1. Из таблиц [1] выбирается неприводимый полином m-й степени и находятся все ненулевые элементы поля $GF(2^m)$. При больших значениях m можно эти вычисления выполнить на ЭВМ.

2. Составляется таблица соответствия входов-выходов.

3. Вычисляются коэффициенты $A(k)$.

Таблица 3

Входы $X = \{X_0, X_1, X_2\}$	Выходы $B(a_j)$	
$0 = 000$	$0 = 000 = 0$	0
$a^0 = 100$	$a^0 = 100 = 1$	0
$a^1 = 010$	$a^0 = 100 = 1$	0
$a^2 = 001$	$a^0 = 100 = 1$	0
$a^3 = 110$	$a^1 = 010 = 0$	1
$a^4 = 011$	$a^1 = 010 = 0$	1
$a^5 = 111$	$a^3 = 110 = 1$	1
$a^6 = 101$	$a^1 = 010 = 0$	1
$a^7 = 100 = a^0$	$a^0 = 100 = 1$	0
	$\downarrow$ S	$\downarrow$ C

4. Выполняется разложение по базисным элементам полученного полиномиального представления функции и приводятся подобные члены с учетом того, что операция сложения выполняется по модулю два.

Пример 1 [21,22]. Рассмотрим ГПФ над полем $GF(2^3)$, образованным с помощью неприводимого полинома $P_3 = X^3 + X + 1$. Полагаем, что $a^0 = 100$, $a^1 = 010$ и $a^2 = 001$ суть линейно независимые элементы поля, а элемент a^1 является его корнем. При этих условиях нетрудно получить остальные четыре элемента: $a^4 = a^3 a^1 = a^2 + a^1 = 011$, $a^5 = a^4 a^1 = a^3 + a^2 = a^2 + a^1 + a^0 = 111$, $a^6 = a^5 a^1 = a^2 = a^0 = 101$ и $a^7 = a^0$. Допустим, что мы хотим построить схему одноразрядного полного сумматора. Рассмотрим таблицу соответствия входов и выходов (табл.3).

Значения X_0, X_1 и X_2 соответствуют первому, второму разряду суммы и переносу соответственно, а S и C обозначают сумму и перенос. Как следует из табл.3, элементы подстановки $B(1) \to B(7)$ в нашем примере — это элементы поля $GF(2^3)$.

Из методических соображений рассмотрим более подробно процесс вычисления коэффициентов $A(k)$:

$$A(1) = \frac{a^0}{a^0} + \frac{a^0}{a^1} + \frac{a^0}{a^2} + \frac{a^1}{a^3} + \frac{a^1}{a^4} + \frac{a^3}{a^5} + \frac{a^1}{a^6} =$$

$$= a^0 + a^0 a^6 + a^0 a^5 + a^1 a^4 + a^1 a^3 + a^3 a^2 + a^1 a^1 = a^0.$$

Операция деления заменена на операцию умножения на инверсный элемент. Далее имеем

$$A(2) = \frac{a^0}{(a^0)^2} + \frac{a^0}{(a^1)^2} + \frac{a^0}{(a^2)^2} + \frac{a^1}{(a^3)^2} + \frac{a^1}{(a^4)^2} + \frac{a^3}{(a^5)^2} + \frac{a^1}{(a^6)^2} = a^1.$$

Аналогичные вычисления дают $A(3) = a^0$, $A(4) = A(7) = 0$, $A(5) = a^4$, $A(6) = 101$. Тогда выражение (3) имеет вид

$$F(X) = X^1 + a^1X^2 + a^4X^5 + a^6X^6 + X^3. \qquad (4)$$

Если рассматривать с точки зрения современной технологии, то равенство (4) можно реализовать двумя способами. 1) Использовать методы табличной арифметики. Однако такая схема была бы слишком громоздкой. 2) Разложить выражение (4) по базисным элементам с целью получения булевых выражений. Для этого достаточно коэффициенты a^4, a^6 и значения переменной X^1, X^2, X^3, X^5 и X^6 представить в виде полиномов. Например, $a^4 = a^1 + a^2$, $a^6 = a^0 + a^2$, $X^2 = X_0 + X_2a^1 + (X_1 + X_2)a^2$ и т.д. Имеем

$$F(X) = (X_0 + X_1a^1 + X_2a^2) + a^1[X_0 + X_2a^1 + (X_1 + X_2)a^2] +$$

$$+ [(X_0 + X_1 + X_2 + X_1X_2)] + [(X_1 + X_0X_1 + X_0X_2)a^1 + (X_2 + X_0X_1)a^2] +$$

$$+ (a^1 + a^2)(X_0 + X_1 + X_2 + X_1X_2) + (X_1 + X_2 + X_0X_2)a^1 +$$

$$+ (X_1 + X_0X_1 + X_0X_2)a^2 + (a^0 + a^2) \times$$

$$\times [(X_0 + X_1 + X_2 + X_1X_2) + (X_2 + X_0X_1)a^1 + (X_1 + X_2X_0)a^2].$$

После умножения и приведения подобных членов получаем окончательно булевы выражения, описывающие работу однозарядового сумматора в логическом базисе И и ИСКЛЮЧАЮЩЕЕ ИЛИ:

$$S = X_0 + X_1 + X_2 \qquad \langle a^0 \rangle$$
$$C = X_0X_1 + X_0X_2 + X_1X_2 \quad \langle a^1 \rangle.$$

Пример 2. Рассчитаем схему последовательного автомата, кодируемого таблицей соответствия, приведенной ниже (см. табл.4). Другими словами, на входы автомата подаются элементы поля $GF(2^4)$ в порядке возрастания их степеней, которые просто генерируются с помощью

счетчика в поле $GF(2^4)$, а на выходах автомата получаются эти же элементы, но в заданном табл 4 порядке. Для построения схемы автомата необходимо вычислить 16 коэффициентов полинома:

$$F(X_0, X_1, X_2 X_3) = B(0) + A(1)X + A(2)X^2 + A(3)X^3 + A(4)X^4 +$$

$$+ A(5)X^5 + A(6)X^6 + A(7)X^7 + A(8)X^8 + A(9)X^9 + A(10)X^{10} +$$

$$+ A(11)X^{11} + A(12)X^{12} + A(13)X^{13} + A(14)X^{14} + A(15)X^{15}. \quad (5)$$

Таблица 4

Входы	Выходы
$X = \{X_0, X_1, X_2, X_3\}$	$F(X)$
0 = 0000	0
a^0 = 1000	a^1
a^1 = 0100	0
a^2 = 0010	a^7
a^3 = 0001	a^5
a^4 = 1100	a^{10}
a^5 = 0110	a^{11}
a^6 = 0011	a^{13}
a^7 = 1101	a^0
a^8 = 1010	a^3
a^9 = 0101	a^{14}
a^{10} = 1110	a^3
a^{11} = 0111	0
a^{12} = 1111	a^8
a^{13} = 1011	a^4
a^{14} = 1001	a^0

Коэффициенты $A(1) \div A(15)$, так же как разложение по базисным элементам и приведение подобных членов, были вычислены с помощью ЭВМ. Получены следующие булевы выражения:

$$
\begin{aligned}
&X_0X_1 + X_2 + X_0X_2 + X_0X_3 + X_1X_2 + X_1X_3 + X_0X_1X_3 + \\
&+ X_1X_2X_3 + X_0X_1X_2X_3 && \langle a^0\rangle \\
&X_0 + X_2 + X_3 + X_1X_3 + X_0X_1X_3 + X_1X_2X_3 && \langle a^1\rangle \\
&X_3 + X_0X_1 + X_0X_3 + X_1X_2 + X_1X_3 + X_1X_2X_3 + X_0X_1X_2X_3 && \langle a^2\rangle \\
&X_2 + X_1X_3 + X_0X_1X_3 + X_0X_2X_3 && \langle a^3\rangle.
\end{aligned}
\qquad (6)
$$

На рис.4 показана принципиальная схема автомата, на рис.5 — принципиальная схема сумматора по модулю, содержащая 144 входа. Наряду с

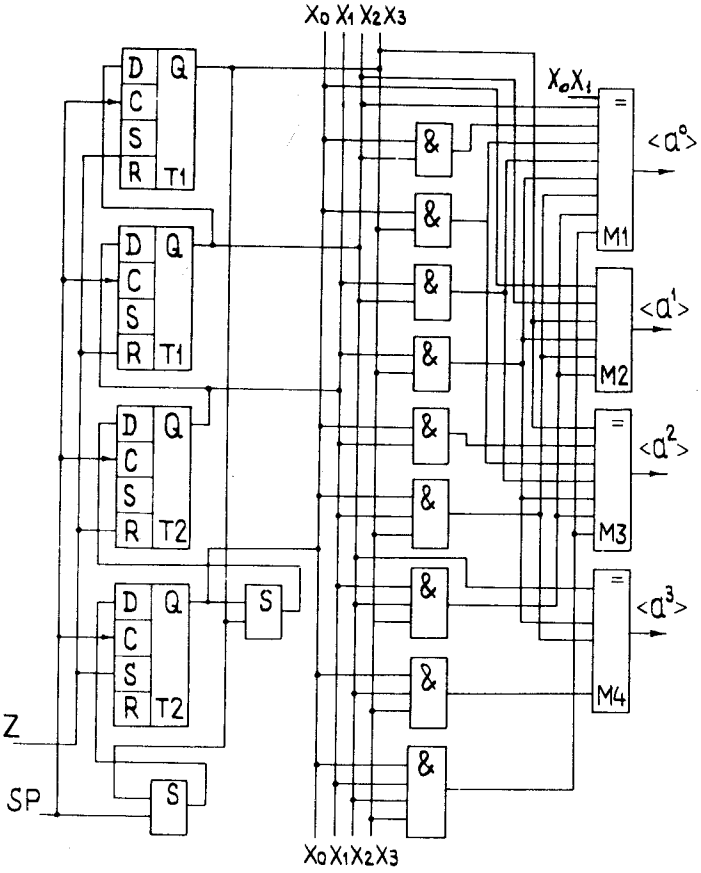


Рис.4. Принципиальная схема последовательностного автомата: $T1$, $T2$, D — триггеры, S — сумматор по модулю 2, M1÷M4 — SN74180 — схемы проверки на четность

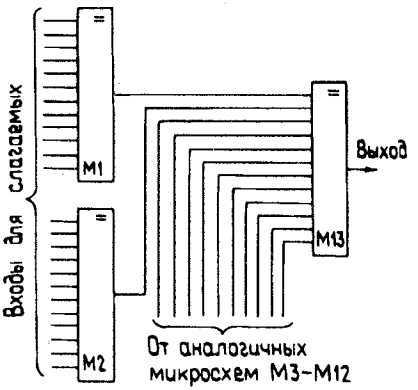


Рис.5. Принципиальная схема сумматора по модулю два на 144 входа. M1÷M13 — микросхемы MC10160

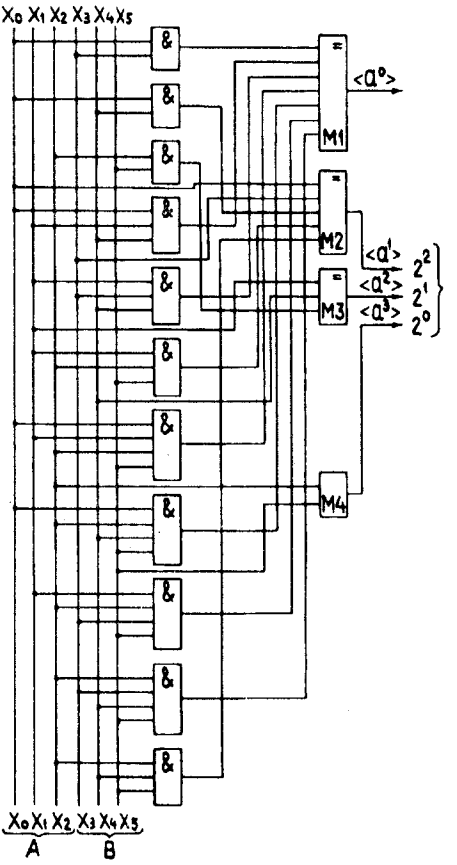


Рис.6. Принципиальная схема трехразрядного сумматора. M1÷M4 — микросхемы SN74180 или MC10160, & — логический элемент И

логическими элементами И такие схемы нужны для быстрого выполнения различного рода операций в поле $GF(2^m)$.

Пример 3. Расчет схемы трехразрядного сумматора с переносом. Данное устройство должно содержать шесть входов и четыре выхода. Это значит, что вычисления необходимо выполнять с ГПФ шести переменных в поле $GF(2^6)$. Причем элементы этого поля рассматриваются как 6-разрядные двоичные числа X_0, X_1, X_2, X_3, X_4, X_5. Среди них первые три цифры представляют первое слагаемое, а последние три — второе слагаемое. С помощью ЭВМ получены следующие булевы выражения:

$$X_0X_3 + X_0X_1X_3 + X_1X_3X_4 + X_0X_1X_2X_5 +$$
$$+ X_0X_2X_4X_5 + X_1X_2X_3X_5 + X_2X_3X_4X_5 \qquad \langle a^0 \rangle \rightarrow 1\text{-й разряд}$$
$$X_0 + X_3 + X_0X_4 + X_1X_2X_5 + X_2X_4X_5 \qquad \langle a^1 \rangle \rightarrow 2\text{-й разряд}$$
$$X_1 + X_4 + X_2X_5 \qquad \langle a^2 \rangle \rightarrow 3\text{-й разряд}$$
$$X_2 + X_5 \qquad \langle a^3 \rangle \rightarrow \text{перенос.}$$

На рис.6 приведена принципиальная схема сумматора. Следует отметить, что таблицу соответствия входов и выходов при таком числе переменных составить вручную трудно. Эту задачу в нашем примере, помимо вычислений, выполняет ЭВМ [28].

КОМБИНИРОВАННЫЕ (СОВМЕЩЕННЫЕ) ОПЕРАЦИИ И ИХ ПРИМЕНЕНИЕ

Анализ выражений (3) и (4) показывает, что для аппаратной реализации ГПФ наряду с такими операциями, как сложение, умножение, деление и возведение в степень, в поле $GF(2^m)$ требуется вычисление сложных выражений, таких как одновременное умножение (деление) степенных членов. Задача заключается в том, чтобы, используя особенности полей Галуа, найти эффективные алгоритмы для аппаратной реализации сложных алгебраических выражений. Известен ряд работ, посвященных параллельным методам выполнения операций в поле Галуа [14,19,29,31]. В свою очередь, автором предложен ряд быстрых алгоритмов, среди которых можно отметить метод совмещенных операций [30,31]. Под совмещенными операциями мы будем подразумевать одновременное вычисление без применения тактовых импульсов, содержащих несколько степенных элементов. Возможны три способа выполнения совмещенных операций.

1. Табличный метод. Использование этого метода основано на том простом факте, что независимо от сложности выражения в результате получается значение одного из элементов поля. С практической точки зрения, это значит, что число адресных входов программируемой памяти определяется только количеством элементов, входящих в данное выражение. Например, емкость памяти, содержащей таблицу решений, будет одинакова при реализации следующих выражений:

$$F_1 = \frac{ABC + D}{DE} \quad \text{и} \quad F_2 = A^p B^q C^r (D^s)^{-1} (E^t)^{-1} + D + C,$$

где A, B, C, D, E, F_1 и F_2 — элементы поля $GF(2^m)$, p, q, r, s и t — целые или дробные числа. Нетрудно заметить, что для вычислений выражений F_1 и F_2 требуется память, имеющая $5m$ входов. Рассмотрим пример применения табличного метода выполнения совмещенных операций.

Синтез универсального динамически программируемого логического модуля. В настоящее время как с теоретической, так, особенно, с практической точки зрения этому направлению уделяется большое внимание. Суть проблемы в том, что усложняются критерии отбора физических событий и, кроме того, появляется необходимость в быстрой перестройке триггерных систем на решение новых задач [29]. Основная проблема заключается в том, чтобы разработать такое логическое устройство, которое выполняло бы экономичным способом и с максимальным быстродействием максимум логических операций при минимальном числе управляющих входов (коэффициентов настройки). Из уравнения (5) следует:

1) Коэффициенты $A(1)+A(15)$ определяют тип реализуемой ГПФ.

2) Для вычисления полинома в целом требуется выполнение таких операций, как сложение, умножение и возведение в степень элементов поля.

Используя метод выполнения совмещенных операций, можно существенно упростить вычисление выражения (5). На рис.7 приведена таблица умножения двух элементов A и B в поле $GF(2^4)$. На рис.8 приведена аналогичная таблица, но при условии, что элемент B к тому же еще возводится в куб. Так, выражение BA^3 для $A = a^7$ и $B = a^{12}$ равно a^3. Или более детально: $a^{12}(a^7)^3 = a^{12}a^{21} = (a^{33}) = a^{15}a^{15}a^3 = a^3$. Таким способом можно составить 14 таблиц, с помощью которых можно программировать содержимое модулей памяти. На рис.9 приведена блок-схема УДПЛМ четырех переменных. Она содержит 4 входа для переменных, 60 входов для коэффициентов настройки и 4 выхода [29]. Как видно из рисунка, для построения такого модуля

B \ A →	a^0	a^1	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}
a^0	a^0	a^1	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}
a^1	a^1	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}	a^0
a^2	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}	a^0	a^1
a^3	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}	a^0	a^1	a^2
a^4	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}	a^0	a^1	a^2	a^3
a^5	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}	a^0	a^1	a^2	a^3	a^4
a^6	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}	a^0	a^1	a^2	a^3	a^4	a^5
a^7	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}	a^0	a^1	a^2	a^3	a^4	a^5	a^6
a^8	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}	a^0	a^1	a^2	a^3	a^4	a^5	a^6	a^7
a^9	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}	a^0	a^1	a^2	a^3	a^4	a^5	a^6	a^7	a^8
a^{10}	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}	a^0	a^1	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9
a^{11}	a^{11}	a^{12}	a^{13}	a^{14}	a^0	a^1	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}
a^{12}	a^{12}	a^{13}	a^{14}	a^0	a^1	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}
a^{13}	a^{13}	a^{14}	a^0	a^1	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}
a^{14}	a^{14}	a^0	a^1	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}

Рис.7. Таблица умножения двух элементов в поле $GF(2^4)$

B \ A →	BA^3	a^0	a^1	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}
	a^0	a^0	a^8	a^6	a^9	a^{12}	a^0	a^3	a^6	a^9	a^{12}	a^0	a^3	a^6	a^9	a^{12}
	a^1	a^1	a^4	a^7	a^{10}	a^{13}	a^1	a^4	a^7	a^{10}	a^{13}	a^1	a^4	a^7	a^{10}	a^{13}
	a^2	a^2	a^5	a^8	a^{11}	a^{14}	a^2	a^5	a^8	a^{11}	a^{14}	a^2	a^5	a^8	a^{11}	a^{14}
	a^3	a^3	a^6	a^9	a^{12}	a^0	a^3	a^6	a^9	a^{12}	a^0	a^3	a^6	a^9	a^{12}	a^0
	a^4	a^4	a^7	a^{10}	a^{13}	a^1	a^4	a^7	a^{10}	a^{13}	a^1	a^4	a^7	a^{10}	a^{13}	a^1
	a^5	a^5	a^8	a^{11}	a^{14}	a^2	a^5	a^8	a^{11}	a^{14}	a^2	a^5	a^8	a^{11}	a^{14}	a^2
	a^6	a^6	a^9	a^{12}	a^0	a^3	a^6	a^9	a^{12}	a^0	a^3	a^6	a^9	a^{12}	a^0	a^3
	a^7	a^7	a^{10}	a^{13}	a^1	a^4	a^7	a^{10}	a^{13}	a^1	a^4	a^7	a^{10}	a^{13}	a^1	a^4
	a^8	a^8	a^{11}	a^{14}	a^2	a^5	a^8	a^{11}	a^{14}	a^2	a^5	a^8	a^{11}	a^{14}	a^2	a^5
	a^9	a^9	a^{12}	a^0	a^3	a^6	a^9	a^{12}	a^0	a^3	a^6	a^9	a^{12}	a^0	a^3	a^6
	a^{10}	a^{10}	a^{13}	a^1	a^4	a^7	a^{10}	a^{13}	a^1	a^4	a^7	a^{10}	a^{13}	a^1	a^4	a^7
	a^{11}	a^{11}	a^{14}	a^2	a^5	a^8	a^{11}	a^{14}	a^2	a^5	a^8	a^{11}	a^{14}	a^2	a^5	a^8
	a^{12}	a^{12}	a^0	a^3	a^6	a^9	a^{12}	a^0	a^3	a^6	a^9	a^{12}	a^0	a^3	a^6	a^9
	a^{13}	a^{13}	a^1	a^4	a^7	a^{10}	a^{13}	a^1	a^4	a^7	a^{10}	a^{13}	a^1	a^4	a^7	a^{10}
	a^{14}	a^{14}	a^2	a^5	a^8	a^{11}	a^{14}	a^2	a^5	a^8	a^{11}	a^{14}	a^2	a^5	a^8	a^{11}

Рис.8. Таблица умножения элемента B на элемент A^3

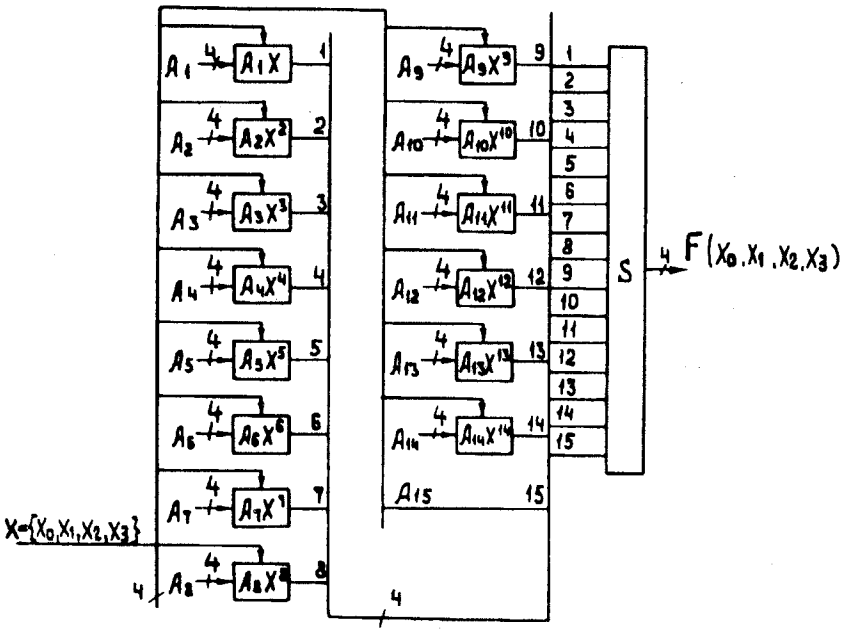


Рис.9. Блок-схема УДПЛМ 4 переменных. S — схема проверки на четность

требуется 15 схем для одновременного умножения с возведением в степень и сумматор по модулю два. Для этих целей можно применить быстрый модуль памяти МС10149. На рис.10 приведена схема для выполнения совмещенных операций. Она состоит из 4-разрядного регистра, в котором хранится значение коэффициента настройки и микросхемы МС10149. Изменяя коэффициенты $A(k)$, где $k = 1, 2, 3, ..., 15$, можно настроить УДПЛМ на выполнение любой из 65536 переключательных функций четырех переменных [29,30].

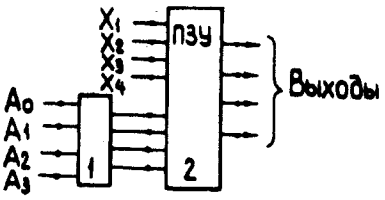


Рис.10. Блок-схема умножения элемента B на степенной элемент A: *1* — регистр, *2* — модуль программируемой памяти МС10149

Рассмотрим несколько простейших логических функций, представленных в табл.5. В первой колонке слева приведены элементы поля

Таблица 5. Представление некоторых логических функций элементами поля $GF(2^4)$

Входы/функция	Совпадение		Повтор		Инверсия	
$X=X_0,X_1,X_2,X_3$	$f(X)=X_0,X_1,X_2,X_3$	$A(k)$	$f(X)=X$	$A(k)$	$f(X)=\overline{X}_0,\overline{X}_1,\overline{X}_2,\overline{X}_3$	$A(k)$
$a^1 = 0100$	0000	a^3	a^1	a^0	1011	a^0
$a^2 = 0010$	0000	a^6	a^2	0000	1101	0000
$a^3 = 0001$	0000	a^9	a^3	0000	1110	0000
$a^4 = 1100$	0000	a^{12}	a^4	0000	0011	0000
$a^5 = 0110$	0000	a^0	a^5	0000	1001	0000
$a^6 = 0011$	0000	a^3	a^6	0000	1100	0000
$a^7 = 1101$	0000	a^6	a^7	0000	0010	0000
$a^8 = 1010$	0000	a^9	a^8	0000	0101	0000
$a^9 = 0101$	0000	a^{12}	a^9	0000	1010	0000
$a^{10} = 1110$	0000	a^0	a^{10}	0000	0001	0000
$a^{11} = 0111$	0000	a^3	a^{11}	0000	1000	0000
$a^{12} = 1111$	1000	a^6	a^{12}	0000	0000	0000
$a^{13} = 1011$	0000	a^9	a^{13}	0000	0100	0000
$a^{14} = 1001$	0000	a^{12}	a^{14}	0000	0110	0000
$a^{15} = a^0 = 1000$	0000	a^0	a^{15}	0000	0000	a^{12}

$GF(2^4)$ и их двоичные эквиваленты. Сигналы, соответствующие этим кодам, подаются на 4 входа модуля. В других колонках приведены соответствующие им значения, получаемые на выходах модуля, и вычисленные с помощью ЭВМ коэффициенты. Полагаем, что ГПФ имеет истинное значение, если она равна единичному элементу $a^0 = 1000$. Для 4-кратных совпадений получены следующие коэффициенты: $A(1) = a^3$, $A(2) = a^6$, $A(3) = a^9$, ..., $A(14) = a^{12}$ и $a(15) = a^0$. В результате получаем уравнение

$$F(X) = a^3X + a^6X^2 + a^9X^3 + a^{12}X^4 + a^0X^5 + a^3X^6 + a^6X^7 +$$
$$+ a^9X^8 + a^{12}X^9 + a^0X^{10} + a^3X^{11} + a^6X^{12} + a^9X^{13} + a^{12}X^{14} + a^0X^{15}. \quad (7)$$

Тот факт, что с помощью уравнения (7) описывается схема совпадений на четыре входа, можно проверить двумя способами.

1. Подставляя $X = a^{12}$, получим $a^3a^{12} = a^0$, $a^6(a^{12})^2 = a^{30} = a^{15}a^{15} = a^0$ и т.д. — 15 одинаковых значений a^0. После сложения по модулю два получим $F(X) = a^0$. Если же в выражение (7) вместо переменной X последовательно подставить остальные значения элементов, то каждый раз получим в выражении (7) 15 различных элементов $a^0 \div a^{14}$, сумма которых равна нулю по определению. Нетрудно заметить, что в таком виде создавать схему совпадений не имеет смысла.

2. Если выражение (7) упростить, разложив элементы поля по базисным элементам, то после упрощения с помощью ЭВМ получим следующее уравнение:

$$F(X) = F(X_0\ X_1\ X_2\ X_3) = a^0.$$

Это значит, что выходы УДПЛМ принимают значение a^0, т.е. истинное, если все входы находятся в состоянии логической единицы.

Рассмотрим вторую колонку в табл.5, с помощью которой иллюстрируется настройка модуля на выполнение «пассивной» операции, когда значения на входах и выходах одинаковы (повтор логических сигналов). В этом случае только один коэффициент не равен нулю. Поэтому имеем

$$F(X) = a^0X = a^0(a^0X_0 + a^1X_1 + a^2X_2 + a^3X_3) = X.$$

Если в уравнение (5) подставить соответствующие коэффициенты a^0 при X и a^{12} при X^{15}, то получим логическое уравнение для инверсии за исключением нулевой точки:

$$F(X) = a^0X + a^{12}X^{15} = X + a^{12}.$$

2. Быстрый алгоритм для выполнения совмещенных операций. Для реализации данного алгоритма требуются логические элементы И и сумматоры по модулю два. Суть метода рассмотрим на примерах. Вначале рассмотрим операцию умножения двух элементов A и B в поле $GF(2^4)$. Имеем

$$P = AB = (A_0a^0 + A_1a^1 + A_2a^2 + A_3a^3)(B_0a^0 + B_1a^1 + B_2^2a + B_3^3a) =$$

$$= P_0a^0 + P_1a^1 + P_2a^2 + P_3a^3,$$

где

$$P_0 = A_0B_0 + A_1B_3 + A_2B_2 + A_3B_1 \qquad \langle a^0\rangle$$
$$P_1 = A_0B_1 + A_1B_0 + A_1B_3 + A_2B_2 + A_2B_3 + A_3B_2 + A_3B_1 \qquad \langle a^1\rangle$$
$$P_2 = A_0B_2 + A_1B_1 + A_2B_0 + A_2B_3 + A_3B_2 + A_3B_3 \qquad \langle a^2\rangle$$
$$P_3 = A_0B_3 + A_1B_2 + A_2B_1 + A_3B_0 + A_3B_3 \qquad \langle a^3\rangle.$$

Полагая $A = B$, получим булевы выражения для возведения какого-либо элемента A в квадрат:

$$P_0^2 = A_0 + A_2 \quad \langle a^0\rangle$$
$$P_1^2 = A_2 \quad \langle a^1\rangle$$
$$P_2^2 = A_1 + A_3 \quad \langle a^2\rangle$$
$$P_3^3 = A_3 \quad \langle a^3\rangle.$$

Очевидно, что для получения выражений, описывающих совмещенную операцию B на A^2, необходимо выполнить следующие вычисления:

$$BA^2=(B_0a^0+B_1a^1+B_2a^2+B_3a^3)[(A_0+A_2)a^0+A_2a^1+(A_1+A_3)a^2+A_3a^3].$$

После упрощения получаем

$$K_0 = B_0A_0 + B_0A_2 + B_2A_1 + B_3A_2 + B_1A_3 + B_2A_3 \qquad \langle a^0\rangle$$
$$K_1 = B_0A_2 + B_1A_2 + B_2A_1 + B_1A_3 + B_3A_2 +$$
$$+ B_3A_1 + B_3A_3 + B_1A_o \qquad \langle a^1\rangle$$
$$K_2 = B_0A_1 + B_0A_3 + B_1A_2 + B_2A_0 + B_2A_2 + B_3A_1 \qquad \langle a^2\rangle$$
$$K_3 = B_0A_3 + B_1A_1 + B_1A_3 + B_2A_2 + B_3A_0 + B_3A_2 + B_3A_3 \qquad \langle a^3\rangle.$$

Пример. Пусть $B = a^3$ и $A = a^{12}$. Тогда $BA^2 = a^3(a^{12})^2 = a^{27} = a^{12}$.

$$B_3 = A_0 = A_1 = A_2 = A_3 = 1 \quad \text{и} \quad B_0 = B_1 = B_2 = 0.$$

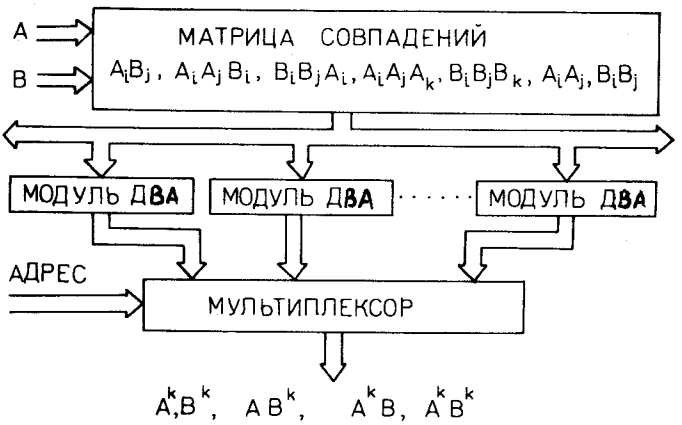


Рис.11. Вариант микросхемы для выполнения совмещенных операций над двумя элементами A и B в поле $GF(2^4)$

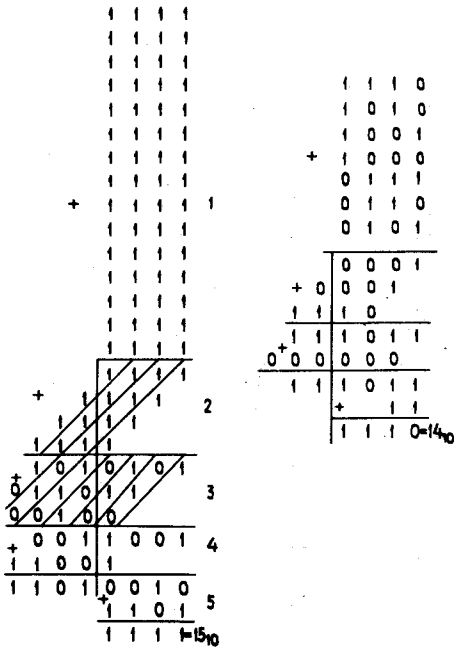
Тогда

$$K_0 = K_2 = K_1 = K_3 = 1 \quad \text{и} \quad BA^2 = a^3(a^{12})^2 = a^{12}.$$

С целью реализации совмещенных операций на базе комбинационных логических схем автором предложен модуль, выполняющий такие процедуры, $B^kA^k(B^k/A^k)$, где $k = 1 - 2^{m-2}$. При $m = 4$ модуль состоит из матрицы логических элементов И, 14 групп сумматоров по модулю два и мультиплексора, выполняющего функции переключения работы модуля на выполнение одной из операций типа BA^2, B^2A, B^2A^3 и т.д. [30] (рис.11). Подобный подход может быть использован и для построения модулей, выполняющих более сложные функции.

3. Применение логарифмов и антилогарифмов. С целью эффективной реализации сложных выражений в поле Галуа Берлекэмп предложил использовать логарифмы и антилогарифмы элементов поля $GF(2^m)$ по основанию a^1 [14, с.71]. Например, в поле $GF(2^4)$ степень произведения двух элементов $A = a^7$ и $B = a^{10}$ равна двум. В самом деле, $\log_a a^7 = 0111_2$ и $\log_a a^{10} = 1010_2$ (младший разряд слева). Складывая полученные двоичные числа, получим 0010_2. Подобные вычисления можно выполнять в случае деления элемента A на элемент B. Логарифмы и антилогарифмы нетрудно получать методами табличной арифметики с помощью модулей программируемой памяти, как это показано в [42].

Рис.12. Два примера, иллюстрирующие работу параллельного циклического компрессора в поле $GF(2^4)$



Используя несложное действие над логарифмами, можно выполнять операцию сложения, заменяя ее на операцию умножения:

$$a^i + a^j = a^i(1 + a^{i-j}).$$

Для того чтобы сократить время циклического суммирования степеней элементов поля при большом числе слагаемых, автором предложен метод циклической компрессии и соответствующее устройство. На рис.12 приведены два примера, иллюстрирующие алгоритм работы циклического компрессора. Первый пример слева соответствует одновременному умножению 15 элементов $a^0 \div a^{14}$ или, что то же самое, возведению в 15 степень элемента a^0. Процесс циклического суммирования (по модулю 15) в данном примере можно разделить условно на пять этапов. На первом этапе в результате подсчета количества единиц в каждом столбце пятнадцать слагаемых сжимаются до четырех, причем результат суммирования записывается по диагонали, начиная с первого столбца справа. Аналогичная процедура выполняется на втором и третьем этапах суммирования. В конце концов из пятнадцати слагаемых получается два, разделенных на две части. Причем, вторая часть суммы представляет собой наибольшее число (11010000), которое равно сумме переносов, возникающих в процессе циклической компрессии. И, наконец, к числу 0010 прибавляем 1101 по модулю 15. На рис.12 справа приведен пример для одновременного вычисления суммы степеней произведения $a^{10}a^{14}a^9a^8a^7a^6a^5 = a^{14}$ в поле $GF(2^4)$. Эти примеры служат одновременно диаграммами для построения принципиальной схемы циклического компрессора (рис.13). На этой схеме не показаны модули программируемой или постоянной памяти (ПЗУ), с помощью которых вычисляются логарифмы и антилогарифмы. Как видно из рис.13, первый каскад компрессора состоит из четырех (15,4) параллельных счетчиков, которые, по су-

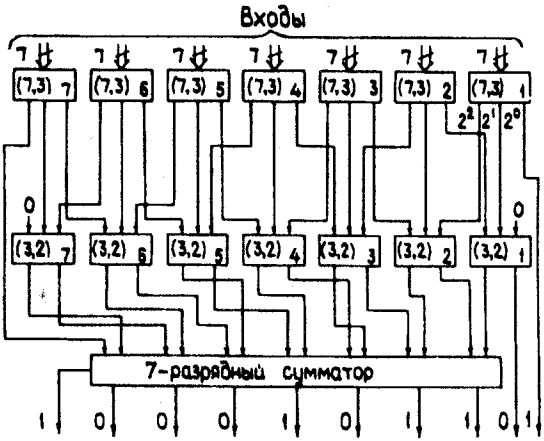


Рис.13. Блок-схема компрессора в поле $GF(2^4)$

ществу, представляют собой дерево полных сумматоров [32]. Второй каскад группы параллельных счетчиков имеет меньшее число входов. После третьего этапа компрессии получается всего два слагаемых, которые суммируются с помощью обыкновенного сумматора с циклическим переносом. Высокое быстродействие работы циклического компрессора обеспечивается за счет применения параллельных счетчиков.

В заключение сделаем некоторые выводы. Используя ГПФ и аналитические вычисления на ЭВМ, можно выполнять синтез сложных логических устройств, в том числе и программируемых модулей, имеющих широкие перспективы применения в триггерных системах первого и второго уровней. В следующих разделах будет показано, каким образом алгебраическая теория кодирования, которая базируется на теории поля Галуа, используется для быстрой компрессии и обработки данных в экспериментах по физике высоких энергий.

МЕТОД СИНДРОМНОГО КОДИРОВАНИЯ

Как известно, в современных многоканальных детекторах заряженных частиц (МДЗЧ) содержится огромное количество позиционно-чувствительных детекторов (каналов передачи). К счастью, во время регистрации события срабатывает лишь ограниченная часть таких детекторов (10—20%), сигналы с которых усиливаются, формируются и обрабатываются с помощью быстрой электроники и специализированных процессоров с целью выработки быстрого триггерного сигнала, запускающего физическую установку. Отбор (фильтрация) полезных событий носит многоуровневый характер. Если рассматривать во времени, то решение на первом уровне должно быть принято в пределах 100 нс, а нередко и менее. Основными критериями отбора событий на первом уровне являются множественность t регистрируемых событий, тип частиц, угол

разлета частиц, наличие вершин распада и т.д. На следующих уровнях могут использоваться более сложные критерии, такие как определение координат событий в условиях большой множественности, вычисление импульсов частиц, вычисление величины энерговыделения с помощью калориметров и пр.

Работа типичной многоуровневой триггерной системы иллюстрируется с помощью табл.6 [33]. Отсчет времени начинается от одной пикосекунды, когда начинается формирование сигналов в детекторах, и вплоть до записи кандидата в полезные события на магнитную ленту.

Таблица 6. Диаграмма, иллюстрирующая процесс обработки события в экспериментах с фиксированной мишенью и на коллайдерах

Фиксированная мишень			Встречные пучки
	1 пс		1 пс
	10 пс		10 пс
	100 пс		100 пс
Первый уровень			
	1 нс	← Частицы проходят кремниевый детектор →	1 нс
		← Запись всех сигналов	
	10 нс		
		Частицы покидают большой детектор →	
Запись всех данных	100 нс		100 нс
		Период встречи пучков. Запись всех сигналов →	
	1 мкс		1 мкс
Второй уровень			
Передача данных в процессор событий	10 мкс	Вычисление p_t*	10 мкс
	100 мкс	Отбор полезных треков и данных о вершинах распада	100 мкс
		Вычисление инвариантных масс частиц	
	1 мс	Начало реконструкции трековых событий	1 мс
Третий уровень	10 мс		10 мс
	100 мс	← Событие записывается на магнитную ленту	100 мс
		Событие записывается на магнитную ленту →	1 с

p_t* — импульс частиц в калориметре.

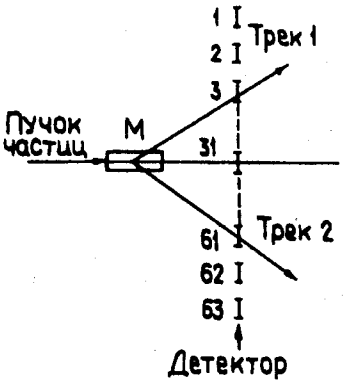


Рис.14. Простая схема, иллюстрирующая метод регистрации двух частиц в сцинтилляционном годоскопе: М — мишень; 1÷63 — сцинтилляторы. Условно показана одна плоскость

На рис.14 приведена простая схема, состоящая из мишени T и детектора. Предположим условно, что детектор состоит из 31 датчика, например, из сцинтилляторов, расположенных в одной плоскости. В результате прохождения двух частиц могут появиться два или более сигналов одновременно. После усиления и формирования они поступают на входы мажоритарной схемы совпадений (МСС) для определения множественности сигналов t и на входы шифратора, с помощью которого определяются координаты частиц (рис.15). Нередко от одной частицы могут сработать два и более сцинтиллятора. В таких случаях (так же, как и в калориметрах) возникает проблема регистрации и идентификации кластеров. Если заведомо известно, что $t = 1$, то построение параллельного шифратора с помощью только комбинационных схем (без элементов памяти) не вызывает затруднений. Для решения проблемы при $t > 1$ часто используются сдвиговые регистры и приоритетные шифраторы. Однако для работы таких схем требуются тактовые импульсы, и поэтому при большом числе каналов регистрации n требуется много времени для определения координат событий. Современные постоянные программируемые запоминающие устройства (ППЗУ) или программируемые логические матрицы (ПЛМ) имеют ограниченное число входов (порядка 15—30) для переменных, и поэтому их применение не может решить проблемы быстрой шифрации при больших значениях $n > 30$. Применение метода синдромного кодирования позволяет ответить на вопрос: как построить быструю МСС и параллельный шифратор с большим числом каналов при $t > 1$ [34]?

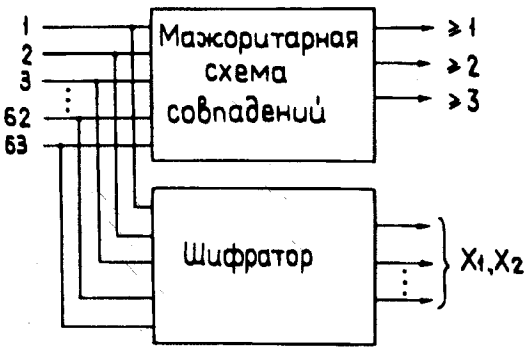


Рис.15. Блок-схема быстрого отбора событий

СИСТЕМА АНАЛОГИЙ

Для применения алгебраической теории кодирования к годоскопическим системам имеет смысл рассмотреть систему аналогий, которая существует между основными выводами и понятиями, используемыми в алгебраической теории кодирования, и алгебраическими методами обработки сигналов в годоскопических системах. В табл.7 приведена такая система аналогий [34].

Таблица 7

Алгебраическая теория кодирования	Алгебраическая теория обработки сигналов в годоскопических системах
1. Избыточное кодирование	Избыточное кодирование с целью повышения пространственного разрешения и увеличение функциональных возможностей годоскопов
2. Кодовый вектор блокового кода, состоящий из n символов	Кодовое слово, считываемое от n позиционно-чувствительных детекторов
3. Вектор ошибки e	Физическое событие, зарегистрированное в многоканальном детекторе
4. Пакет ошибок	Событие с кластером
5. Корректирующая способность кода t	Количество одновременно сработавших позиционно-чувствительных детекторов t
6. Число проверочных символов mt	Количество разрядов на выходах параллельного шифратора
7. Эффективность кода	Коэффициент сжатия K_c
8. Кодирующее устройство	Параллельный шифратор
9. Проверочная матрица	Кодирующая матрица
10. Кодовое расстояние d	Кодовое расстояние d
11. Вес кодового вектора	Вес строки кодирующей матрицы
12. Вес столбца проверочной матрицы	Коэффициент разветвления сигнала датчика K_p
13. Полностью асимметричный канал	Канал регистрации в годоскопической плоскости
14. Итеративное кодирование	Кодирование данных в детекторе, состоящем из нескольких годоскопических плоскостей

Прокомментируем табл.7 по пунктам.

1. Избыточное кодирование используется в технике связи и в вычислительной технике для повышения надежности устройств передачи и обработки информации. В этом смысле избыточное кодирование

широко применяется в электронных методах физики высоких энергий. Однако, как показано в работах автора, техника корректирующих кодов может быть успешно использована и для создания детекторов и годоскопических систем, имеющих оптимальное соотношение между множественностью регистрируемых сигналов, пространственным разрешением и простотой кодирующей схемы.

2. Блоковый код — это корректирующий код, в котором используется последовательность из n символов. Кодовый вектор блокового кода состоит из k информационных и $m = n - k$ избыточных контрольных разрядов. Вектор, состоящий из одних нулей, называется нулевым вектором. Нулевому вектору соответствует нулевое слово, считываемое от детектора или годоскопической плоскости в том случае, когда нет сработавших датчиков. В отличие от теории кодирования, где кодовое слово чаще всего рассматривается как обыкновенный двоичный код, передаваемый по каналу связи, в годоскопических системах информация считывается от позиционно-чувствительных датчиков и поэтому координаты сработавших датчиков получаются в виде унитарного позиционного кода.

3. В процессе передачи данных по каналу к кодовому вектору может добавиться вектор ошибки e. Этому вектору в теории годоскопических систем соответствует физическое событие, в результате которого с датчиков поступают сигналы по каналам регистрации.

4. Пакету ошибок в каналах передачи соответствует кластер, который возникает в результате срабатывания группы соседних датчиков. Поэтому теория кодов, исправляющих пакеты ошибок, может быть применена для построения устройств, способных регистрировать координаты кластеров.

5. Параметр t есть то количество искаженных информационных символов, которое может исправляться данным кодом. В теории годоскопических систем значение t определяет максимальное число сработавших датчиков, координаты которых однозначно определяются координатным процессором.

6. Важным параметром кода является число проверочных символов (синдром кода). Эта величина зависит от конструкции кода, блоковой длины, и от параметра t. Так, если взять широко известный код Хэмминга, у которого $t = 1$, то блоковая длина $n = 2^m - 1$, а число информационных символов $k = n - m$. Если же $t > 1$, то для оптимальных кодов число проверочных символов $N = mt$.

7. Эффективность кода v определяется из отношения $v = k/n$. В годоскопических системах этому параметру соответствует коэффициент сжатия K_c, равный n/N. Это один из важнейших параметров годоскопа, в

котором используется синдромное кодирование, так как он характеризует степень сжатия данных.

8. Кодирующее устройство служит для формирования контрольных символов на стороне передатчика. В годоскопических системах аналогом кодирующего устройства служит параллельный шифратор. В сцинтилляционных годоскопах кодирование может выполняться на уровне конструкции годоскопа, например, путем выполнения соответствующих связей сцинтилляторов с ФЭУ при помощи световодов. Следует отметить, что кодирующее устройство может быть как параллельного, так и последовательного действия на основе сдвиговых регистров с логической обратной связью.

9. Структура кодирующего устройства задается матрицей проверочных соотношений, состоящей из нулей и единиц и содержащей N строк и n столбцов. Примеры таких матриц будут довольно часто встречаться в обзоре. В соответствии с принятой аналогией для построения параллельных шифраторов с заданными свойствами можно использовать проверочные матрицы, которые, применительно к годоскопическим системам, мы будем называть матрицами связей или кодирующими матрицами. По матрицам связей можно определять не только основные параметры параллельных шифраторов, но и их принципиальные схемы. С целью исключения неопределенностей при вычислении координат сработавших датчиков, и даже одного из них, все столбцы матрицы связей должны быть различны, и каждому позиционно-чувствительному датчику ставится в соответствие один столбец. С целью анализа функциональных возможностей параллельного шифратора над столбцами соответствующей ему матрицы связей выполняются линейные и нелинейные операции. Так, если всевозможные суммы двух столбцов различны, то с помощью такой матрицы можно построить параллельный шифратор для определения координат по крайней мере двух сработавших позиционно-чувствительных датчиков. Далее, количеству строк матрицы связей соответствует число выходов N параллельного шифратора.

10. Одним из важнейших параметров корректирующего кода является его кодовое расстояние d. Зная кодовое расстояние того или иного кода, можно выбрать или составить кодирующую матрицу с нужными свойствами. Поскольку исходное кодовое слово, считываемое от детектора, рассматривается как нулевое, то нередко более удобно пользоваться следующей теоремой [65]: линейный (n,k)-код с проверочной матрицей $H = [h_0, h_1, h_{n-1}]$, где h_i, $i = 0, 1, ..., n-1$ — векторы-столбцы размерности $[n-k]\times 1$, имеет минимальное кодовое расстояние d тогда и только тогда, когда любые $d-1$ столбцов кодирующей матрицы $H_{n,N}$ линейно независимы. Пользуясь данной теоремой, можно использовать ЭВМ

для расчета кодирующих матриц с заданными свойствами, особенно при больших значениях n.

11. Вес кодового вектора w определяется как число ненулевых компонентов этого вектора. Количество единиц в строке проверочной матрицы, так же, как и в строке матрицы связей, характеризует сложность реализации кодирующих устройств, так как число входов схемы проверки на четность равно числу единиц в строках кодирующей матрицы.

12. Вес столбца проверочной матрицы также имеет отношение к сложности реализации кодирующего устройства или параллельного шифратора. Так, вес столбца матрицы связей определяет коэффициент разветвления сигнала K_p. Чем меньше эта величина при прочих равных параметрах, тем проще связи между выходами датчиков и логическими элементами или усилителями параллельного шифратора.

13. Полностью асимметричным является канал, в котором имеет место только один вид ошибок, т.е. возможно либо преобразование нулей в единицы, либо, наоборот, только единиц в нули. В этом смысле каналы считывания данных в годоскопических системах являются чисто асимметричными. Необходимость введения такой аналогии диктуется тем, что для асимметричных каналов имеется ряд кодов с хорошими параметрами, и такие коды легко реализуются.

14. Простейшим итеративным кодом является двумерный итеративный код, который используется для оптимального кодирования данных как в сцинтилляционных годоскопах, так и в МПК. При таком способе кодирования n датчиков располагаются в виде квадратной матрицы, содержащей k строк и k столбцов, и вычисляется синдром по строкам и столбцам. В результате число усилителей-смесителей сигналов (ФЭУ) уменьшается до $2n^{1/2}$. Причем синдром вычисляется с помощью усилителей-смесителей. Следует отметить, что в двумерном итеративном коде общее кодовое расстояние $d = d_1 d_2$, где d_1 и d_2 — кодовые расстояния исходных кодов, выбранных для итерации. Важно, что итеративные коды образуют обширный и интересный с практической точки зрения класс кодов, поскольку для итерации можно выбирать любые коды с хорошими исходными параметрами.

Были рассмотрены основные аналогии и, вообще говоря, их можно было бы продолжить. Например, весьма важные с практической точки зрения выводы можно сделать из следующей аналогии. В теории кодирования известен редко используемый на практике так называемый суперпозиционный код, содержащий M кодовых слов, таких, что для положительного числа g булева сумма g различных кодовых слов отличается от каждой суммы g или меньшего числа кодовых слов. Кроме того, эти суммы должны отличаться от слагаемых. Например:

$$\begin{array}{ccc} 1000 & 1101 & 0110 \\ \vee & \vee & \vee \\ 0100 & 0011 & 0001 \\ \hline 1100 & 1111 & 0111 \end{array}$$

Практически это значит, что шифраторы для суперпозиционных кодов могут создаваться на основе смесителей сигналов (ФЭУ), обычных усилителей-смесителей, волоконно-оптических световодов и т.д. Однако при прочих равных параметрах суперпозиционные коды имеют меньшее кодовое расстояние и соответственно величину коэффициента сжатия K_c, так как формирование синдрома выполняется не по модулю два, а по правилам булевой суммы, где число получаемых кодовых комбинаций меньше. Например: $1 + 1 = 0$; $1 + 1 + 1 + 1 = 0$; $1 + 0 = 1$; $0 + 1 = 1$ (mod 2). В то же время, складывая по правилам булевой суммы, имеем: $1 + 1 = 1$; $1 + 1 + 1 + 1 = 1$; $0 + 1 = 1$; $1 + 0 = 1$.

Таким образом, суть метода синдромного кодирования заключается в следующем (рис.16). Вначале рассмотрим типичную многоразрядную систему передачи, где используются корректирующие коды (рис.16,*а*), которая содержит на стороне передатчика одноразрядный регистр. В процессе передачи и кодирования к одноразрядному слову в соответствии с выбранным кодом добавляется k-разрядный код синдрома. Затем $(1+k)$-разрядное слово передается в приемник, где расположено декодирующее устройство. Если в процессе передачи по каналу связи возникли ошибки, то они исправляются (в определенных пределах) и затем одноразрядное слово запоминается на регистре. Теперь рассмотрим более простую схему передачи, которая имеет место в эксперименте (рис.16,*б*). В отсутствие события или ложно сработавших позиционно-чувствительных детекторов информационное слово всегда равно нулю, а регистрация события рассматривается как добавление вектора ошибки к нулевому информационному слову, длина которого равна количеству каналов регистрации n.

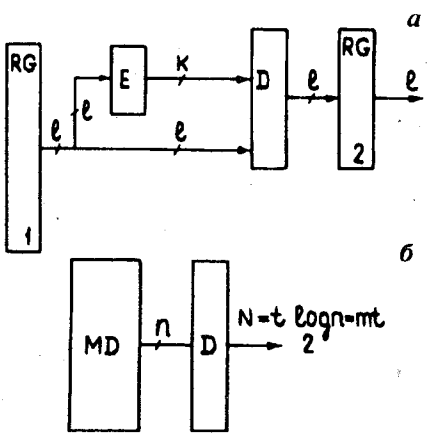


Рис.16. Пояснение к методу синдромного кодирования. *а)* Обычная система передачи, где используются корректирующие коды. К — шифратор, Д — дешифратор. *б)* Многоканальная система передачи, в которой используется метод синдромного кодирования. МД — многоканальный детектор

Если использовать оптимальный код, исправляющий t ошибок, то на выходе параллельного шифратора длина считываемого слова сжимается до $N = \log_2 n$. Например, число каналов регистрации равно 31, а сигналы поступили от 10 и 22 каналов передачи:

$$0000000001000000000001000000000$$

(счет каналов ведется слева направо). Если выбрать БЧХ-код (код Боуза — Чаудхури — Хоквингема), исправляющий $t = 2$ ошибок, то на выходе кодирующего устройства (шифратора) получается синдром N, равный

$$N = \log_2 31 = 10.$$

Здесь необходимо сделать следующие выводы.

При $t << n$ (это условие обычно выполняется на практике) эффективность метода синдромного кодирования увеличивается с ростом n. В случае использования БХЧ-кода на выходах кодирующего устройства получается код, эквивалентный определенному количеству элементов поля Галуа $GF(2^m)$, над которыми можно выполнять различного рода арифметические и алгебраические операции.

Код синдрома несет в себе данные как о числе сработавших позиционно-чувствительных детекторов, так и об их координатах.

Таким образом, использование метода синдромного кодирования предполагает следующие процедуры: 1) компрессия данных с помощью параллельных шифраторов t; 2) вычисление множественности сигналов; 3) определение координат сработавших позиционно-чувствительных детекторов (датчиков) в годоскопической плоскости детектора.

ПАРАЛЛЕЛЬНЫЕ ШИФРАТОРЫ

В соответствии с методом синдромного кодирования датчики детектора нумеруются в порядке возрастания степеней элементов поля $GF(2^m)$. Преобразование унитарного кода, считываемого с многоканального детектора, в элементы поля после усиления и формирования сигналов выполняется с помощью кодирующего устройства, которое, как это принято в теории БЧХ-кодов, описывается с помощью матрицы проверочных соотношений [34—36] H^T (кодирующей матрицы $H_{n,mt}$). Подробное описание теории БЧХ-кода можно найти в [1]. В общем виде кодирующая матрица $H_{n,mt}$ имеет вид

$$H_{n,mt} = \begin{vmatrix} 1 & 1 & 1 & 1 \\ a^1 & a^3 & a^5 & a^{2t-1} \\ a^2 & a^6 & a^{10} & a^{2(2t-1)} \\ a^3 & a^9 & a^{15} & a^{3(2t-1)} \\ \cdots & \cdots & \cdots & \cdots \\ a^{n-1} & a^{3(n-1)} & a^{5(n-1)} & a^{(n-1)a(2t-1)} \end{vmatrix}. \qquad (8)$$

Видно, что в первой колонке содержатся элементы поля $GF(2^m)$ в порядке возрастания их степеней. Во второй колонке расположены кубы соответствующих им элементов первой колонки и т.д. С целью упрощения положим, что многоканальный детектор имеет $n = 2^6 - 1 = 63$ датчика ($m = 6$). Это значит, что поле Галуа образуется с помощью полинома $X^6 + X + 1$. Для множественности $t \le 4$ кодирующая матрица $H_{63,24}$ имеет вид:

Позиции датчиков						Двоичные эквиваленты			
0 *	1	1	1	1		100000	100000	100000	100000
1	a^1	a^3	a^5	a^7		010000	000100	000001	001100
2	a^2	a^6	a^{10}	a^{14}		001000	110000	000011	001010
3 *	a^3	a^9	a^{15}	a^{21}		000100	000110	000101	110111
4 $H_{63,24}$=	a^4	a^{12}	a^{20}	a^{28}	→	000010	101000	001111	001110
5 *	a^5	a^{15}	a^{25}	a^{35}		000001	000101	010001	110100
⋮	.....	.....	.....	.....		.....	.....	.....	.....
59	a^{59}	a^{51}	a^{43}	a^{35}		101111	110101	111011	110100
60	a^{60}	a^{54}	a^{48}	a^{42}		100111	111010	101100	010111
61	a^{61}	a^{57}	a^{53}	a^{49}		100011	011111	010101	010110
62	a^{62}	a^{60}	a^{58}	a^{56}		100001	100111	111111	111110
	↓	↓	↓	↓		↓	↓	↓	↓
	S_1	S_3	S_5	S_7		S_1	S_3	S_5	S_7

Допустим, что одновременно сработали датчики на позициях $X_1 = a^0$, $X_2 = a^3$ и $X_3 = a^5$. Тогда для вычисления степенных симметрических функций Ньютона $S_1 \div S_7$ необходимо сложить по модулю два элементы, расположенные на позициях, обозначенных *. Имеем

$$
\begin{aligned}
S_1 &= a^0 + a^3 + a^5 = a^{23},\\
S_3 &= a^0 + a^9 + a^{15} = a^{61},\\
S_5 &= a^0 + a^{15} + a^{25} = a^{35},\\
S_7 &= a^0 + a^{21} + a^{35} = a^{61}.
\end{aligned}
\qquad (9)
$$

Для получения высокого быстродействия синдром вычисляется с помощью параллельных схем проверки (рис.17). Анализ матрицы $H_{63,24}$ показывает, что количество схем проверки на четность можно существенно уменьшить, если сгруппировать совпадающие единицы в соответствующих колонках матрицы проверочных соотношений [5].

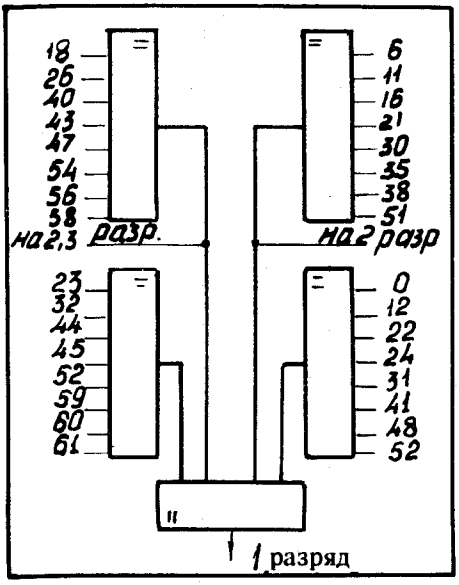


Рис.17. Принципиальная схема для вычисления одного разряда синдрома. Микросхемы — MC10160

Формирование кода синдрома можно выполнить всего за 12 нс, если для этих целей использовать микросхему MC10160 с задержкой 6 нс. В нашем примере число разрядов синдрома $N = 18$ для $n = 63$ и $t = 3$. Таким образом, унитарный 63-разрядный код трансформирован в 18-разрядный код, состоящий из трех элементов поля $GF(2^6)$ S_1, S_3 и S_5. При этом коэффициент сжатия $K_c = 63/18$. Значение S_7 будет использовано в примерах, приведенных ниже.

НОВЫЙ ТИП МАЖОРИТАРНЫХ СХЕМ СОВПАДЕНИЙ

Поскольку код синдрома БЧХ-кода несет в себе данные о величине множественности t, то следующим этапом после компрессии данных является определение величины t. С этой целью используется следующее свойство матрицы L_t. Матрица [1,37] размерности $t \times t$:

$$L_t = \begin{vmatrix} S_1 & 1 & 0 & 0 & 0.....0 \\ S_3 & S_1^2 & S_1 & 1 & 0.....0 \\ S_5 & S_1^4 & S_3 & S_1^2 & S_1.....0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ S_{2t-1} & S_{2t-2} & S_{2t-3} & S_{2t-4} & S_{2t-5}..S_t \end{vmatrix} \qquad (10)$$

невырождена, если степенные симметрические функции зависят от t или $t+1$ элементов поля, и вырождена, если S_j зависят от меньшего, чем $t-1$, числа различных элементов поля. Другими словами, для вычисления величины t необходимо вычислять детерминант $\det L_t$ в поле Галуа. Выражения для детерминантов при $t = 1{+}4$ приведены в табл.8.

Таблица 8

t	$\det L_t$
1	S_1
2	$S_1^3 + S_3$
3	$S_1^6 + S_1^3 S_3 + S_1 S_5 + S_3^2$
4	$S_1^{10} + S_1^7 S_3 + S_1^5 S_5 + S_1^2 S_3 S_5 + S_1 S_3^3 + S_3 S_7 + S_5^2 + S_1^3 S_7$

Очевидно, что значение $\det L_t$ может быть равно нулю или значению какого-либо поля. Поэтому логические выражения для мажоритарной схемы совпадений (МСС), содержащей n входов и t выходов, имеют вид [7]:

$$\begin{aligned} &\text{выход1} = \det L_1 \vee \det L_2 \vee \det L_3 \vee \ldots \det L_j \vee \ldots \det L_t \geq 1 \\ &\text{выход2} = \det L_2 \vee \det L_3 \vee \ldots \det L_j \vee \ldots \det L_t \geq 2 \\ &\text{выход3} = \det L_3 \ldots \det L_j \vee \det L_t \geq 3 \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ &\text{выход } j = \det L_j \vee \det L_t \geq j \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ &\text{выход } t = \det L_t \geq t. \end{aligned} \qquad (11)$$

Если к МСС, описываемой равенствами (11), добавить несложные схемы анализа детерминантов на «0» и «1», можно получить также строгие равенства, как это показано на рис.18. Следует отметить, что для практической реализации таких устройств можно использовать как табличные методы, так и другие алгоритмы, рассмотренные выше. Схе-

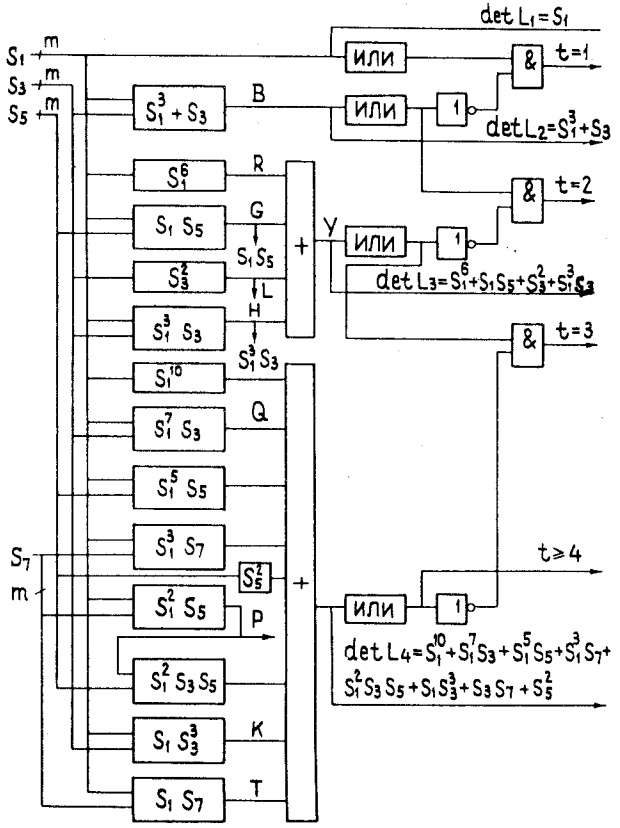


Рис.18. Блок-схема вычисления детерминантов в поле $GF(2^m)$ для $t = 1 \div 4$

ма, изображенная на рис.18, разработана таким образом, чтобы для ее реализации при больших числах можно было использовать ППЗУ, содержащие $2m$ адресных входов. При $t = 2$ или 3 и $m = 10 \div 15$ можно применить программируемые логические матрицы (ПЛМ).

Рассмотрим пример. Пусть $t = 3$. Имеем

$$\det L_1 = S_1 = a^{23} \neq 0,$$

$$\det L_2 = a^{69} + a^{61} = a^6 + a^{61} = a^{46} \neq 0,$$

$$\det L_3 = a^{138} + a^{69}a^{61} + a^{23}a^{35} + a^{122} = a^{12} + a^4 + a^{58} + a^{59} \neq 0, \quad (12)$$

но

$$\det L_4 = a^{41} + a^{33} + a^{24} + a^{16} + a^{17} + a^{59} + a^7 + a^4 = 0.$$

Таким образом, анализируя значения определителей, можно быстро вычислить значения t.

ОПРЕДЕЛЕНИЕ КООРДИНАТ СОБЫТИЙ

Используя свойства синдрома и алгебраические методы декодирования позиций ошибок, одновременно с определением множественности t можно определять координаты X_i ($i = 1, 2, 3, ..., t$) событий. Здесь следует выделить два момента.

1) При $t \leq 5$ можно использовать табличные методы.

2) В общем виде для нахождения координат событий необходимо решить уравнение [1], которое названо координатным [6]:

$$P(X) = X^t + \sigma_1 X^{t-1} + \sigma_2 X^{t-2} + \ldots + \sigma_t = \\ = (X + X_1)(X + X_2) \ldots (X + X_j)(X + X_t). \qquad (13)$$

Между степенными симметрическими функциями Ньютона S_j и координатами событий X_i имеют место соотношения (для двоичного случая)

$$S_j = \sum_{i=1}^{t} X_i^j, \quad i = 1, 2, 3, \ldots t; \quad j = 1, 3, 5, \ldots (2t - 1).$$

Поэтому алгоритм Питерсона (применительно к методу синдромного кодирования) для определения координат событий состоит из трех этапов.

1) Вычисление значений S_j.

2) Вычисление степенных симметричных функций σ_i. В свою очередь, значения σ_t связаны S_j-соотношениями Ньютона (для двоичной арифметики):

$$S_1 + \sigma_1 = 0,$$

$$S_3 + \sigma_1 S_1^2 + \sigma_2 S_1 + \sigma_3 = 0,$$

$$S_5 + \sigma_1 S_1^4 + \sigma_2 S_3 + \sigma_3 S_1^2 + \sigma_4 S_1 + \sigma_5 = 0,$$

.....

.....

3) Нахождение корней X_i полинома $P(X)$. Корни находятся методом последовательных подстановок всевозможных элементов поля в уравнение (13). Нетрудно заметить, что такой метод определения координат

событий требует много времени. Однако при $t \le 5$ известны табличные методы вычисления корней уравнения (13) [38—44]. Более детально рассмотрим два случая: $t = 2$ и $t = 3$.

При $t = 2$ имеем

$$X^2 + \sigma_1 S + \sigma_2 = 0, \tag{14}$$

где $S_1 = \sigma_1$ и $\sigma_2 = (S_1^3 + S_3)/S_1$. Подставляя $X = \sigma_1 Y$ в уравнение (14), получим

$$Y^2 + Y = \gamma, \tag{15}$$

где $\gamma = \sigma_2/\sigma_1^2$, $X_1 = \sigma_1 Y_1$, $X_2 = \sigma_1 Y_2$ и $Y_2 = Y_1 + 1$.

Алгоритм решения квадратного уравнения [14, с.252] удачно приспособлен для его реализации методом совмещенных операций. Показано, что если оператор $T(r) = 0$, то

$$Y_1 = \sum_{i=0}^{m-1} \gamma_i y_i . \tag{16}$$

В свою очередь, значения y_i определяются из соотношения

$$y_i^2 + y = \begin{cases} a^i & \mathrm{Tr}(a^i) = 0 \\ a^i + a^k = 1. \end{cases}$$

После несложных вычислений получаем следующие значения $y_0 \div y_5$ для $m = 6$ [6]:

$$y_0 = a^0 \text{ для } a^i = 0, \quad y_1 = a^{11} \text{ для } a^i = a^{36}, \quad y_2 = a^{55} \text{ для } a^i = a^{32},$$

$$y_3 = a^0 \text{ для } a^i = 0, \quad y_4 = a^{33} \text{ для } a^i = a^{38}, \quad y_5 = a^{43} \text{ для } a^i = a^{19}.$$

В поле $GF(2^6)$

$$\begin{gathered} y_1 = y_{10}a^0 + y_{11}a^1 + y_{12}a^2 + y_{13}a^3 + y_{14}a^4 + y_{15}a^5 \\ \text{и } \gamma = \gamma_0 a^0 + \gamma_1 a^1 + \gamma_2 a^2 + \gamma_3 a^3 + \gamma_4 a^4 + \gamma_5 a^5. \end{gathered} \tag{17}$$

Поэтому из (16) имеем

$$y_{10} = \gamma_0, \quad y_{11} = \gamma_0 + \gamma_1 + \gamma_5, \quad y_{12} = \gamma_1 + \gamma_2 + \gamma_3 + \gamma_5, \quad y_{13} = \gamma_0,$$

$$y_{14} = \gamma_0 + \gamma_3 + \gamma_5, \quad y_{15} = \gamma_0 + \gamma_1 + \gamma_2 + \gamma_4 + \gamma_5.$$

На рис.19 изображена блок-схема решения уравнения (14) [43]. Быстродействие координатного процессора можно оценить из равенства

$$T_{k2} = T_y + 2T_s + T_{\text{ППЗУ}\gamma},$$

где T_y — время умножения двух элементов в поле $GF(2^m)$, T_s — время сложения по модулю два и $T_{\text{ППЗУ}}$ — задержка в ППЗУ. Если применить быстродействующие логические схемы, то можно вычислить одновременно две координаты менее чем за 10 нс. Рассмотрим конкретный пример. Допустим, что одновременно сработали датчики на позициях $X_1=a^0$ и $X_2=a^2$. Имеем $S_1 = a^{12}$, $S_3 = a^1$, $\sigma_1=a^{12}$, $\sigma_2=a^2$ и $\gamma = a^{41} = 101110$. Из уравнений (17) получаем $y_0= = y_1 = y_3 = y_5 = 1$, $y_0 = y_4 = 0$. Тогда $Y_1 a^{51}$ и $Y_2 = a^{53}$. И, наконец,

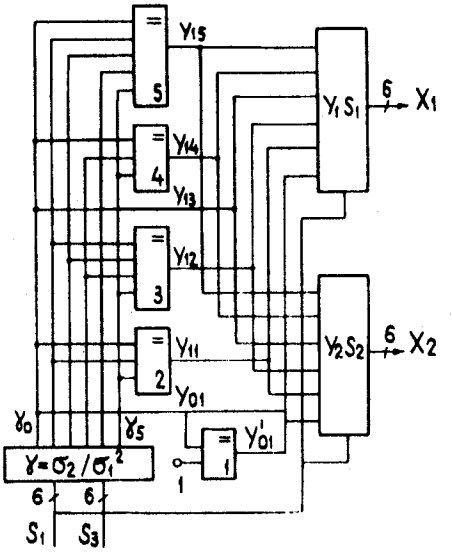


Рис.19. Блок-схема решения координатного уравнения второй степени в $GF(2^m)$

$$X_1 = a^{12}a^{51} = a^0 \text{ и } X_2 = a^{12}a^{53} = a^2.$$

Можно проверить, что значения элементов a^0 и a^2 удовлетворяют уравнению

$$X^2 + a^{12}X + a^2 = 0.$$

Решение кубического уравнения ($t = 3$). При $t = 3$ разрядность синдрома равна $3m$. Если величина $3m$ такова, что прямое декодирование невозможно из-за ограниченной емкости ППЗУ или ПЛА, то можно применить параллельно-последовательный, более экономичный табличный способ решения кубического уравнения. Имеем

$$X^3 + \sigma_1 X^2 + \sigma_3 = 0. \tag{18}$$

После подстановок $X = \sigma_1 + Y$ и $Y = Z(\sigma_1^2 + \sigma_2)^{1/2}$ уравнение (18) приводится к виду [40.41]

$$Z^3 + Z = C, \tag{19}$$

где

$$C = \frac{\sigma_1\sigma_2 + \sigma_3}{(\sigma_1^2 + \sigma_2)(\sigma_1^2 + \sigma_2)^{1/2}}. \tag{20}$$

С целью упрощения вычисления константы C имеет смысл выразить ее непосредственно через S_1, S_3 и S_5. Имеем

$$\frac{\left(\dfrac{S_3^{\,2}+S_1^{\,6}}{S_1^{\,3}+S_3}\right)}{\left(\dfrac{S_1^{\,5}+S_5}{S_1^{\,3}+S_3}\right)\left(\dfrac{S_1^{\,5}+S_5}{S_1^{\,3}+S_3}\right)^{1/2}} = D(E\,E^{1/2})^{-1}.$$

Блок-схема решения кубического уравнения табличным способом приведена на рис.20 [43]. Время решения T_{k3} можно вычислить из соотношения

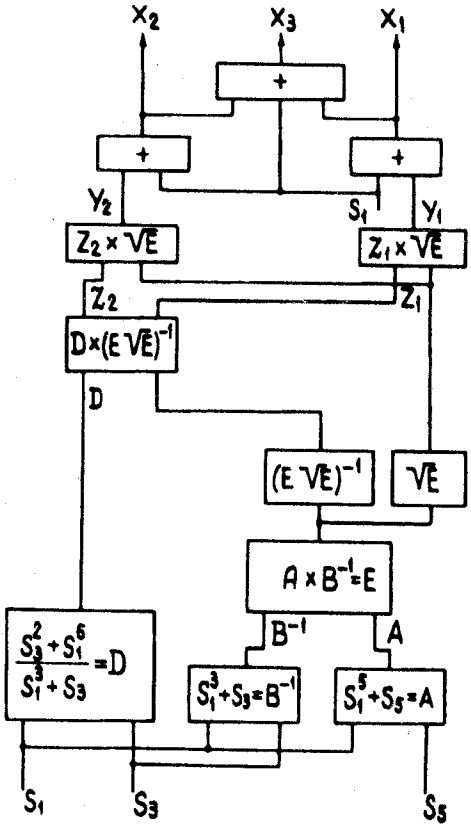


Рис.20. Блок-схема решения координатного уравнения третьей степени в поле $GF(2^m)$

$$T_{k3} = 5T_{\text{ППЗУ}} + 2T_s\,.$$

В работе [41] приведен алгоритм решения уравнения четвертой степени

$$\sigma(X) = X^4 + \sigma_1 X^3 + \\ + \sigma_2 X^2 + \sigma_3 X + \sigma_4. \qquad (21)$$

Это уравнение после ряда подстановок сводится к решению двух квадратных уравнений, которые затем решаются табличным способом, как это описано выше. В [42] решается такая проблема, но операции выполняются над логарифмами элементов поля. Блок-схема координатного процессора для $t=4$ и $t=5$ описана в [44]. При этом с целью экономичной реализации значений σ_t автор использовал метод трансформации Берликэмпа [45]:

$$\sigma_3 = R_3 + R_1\sigma_2, \qquad (22)$$

$$\sigma_4 = \frac{R_7 + R_5\sigma_2}{R_3}, \qquad (23)$$

где

$$\sigma_1 = R_1$$

$$\sigma_3 = R_3 + R_1\sigma_2$$

$$R_5 + R_3\sigma_2 + R_1\sigma_4 = 0$$

$$R_7 + R_5\sigma_2 + R_3\sigma_4 = 0 \qquad (24)$$

$$R_3 = S_3 + S_1^3 = B$$

$$R_5 = S_5 + S_1^2 S_3 = V$$

$$R_7 = S_7 + P + V + S_1^7,$$

где $P = S_1^2 S_5$ (рис.18). Уравнение (24) не содержит R_7 и поэтому вычисляется проще, чем (23):

$$\sigma_4 = \frac{R_5 + R_3\sigma_2}{R_1}. \qquad (25)$$

Поэтому после предварительного вычисления σ_2, которое вычисляется относительно просто, из (22) и (23) находим σ_3 и σ_4. Формулы для вычисления σ_2 при $t = 2{+}5$ даны в [46]. Решение уравнения пятой степени сводится к решению квадратного и кубического уравнений [41,42,44].

ГИБРИДНЫЙ МЕТОД РЕШЕНИЯ КООРДИНАТНОГО УРАВНЕНИЯ

Сложность вычислений значений σ_t и S_j существенно возрастает с ростом множественности t. Поэтому определенный интерес представляет метод решения координатного уравнения в общем виде, без вычисления значений σ_t [47]. Такой подход позволяет, используя свойства синдрома, разработать гибридный процессор. С помощью такого процессора можно в зависимости от значения t решать координатное уравнение или табличным методом, если $t \le 5$, или методом циклических сдвигов полученного синдрома, в котором используются следующие свойства определителей A и d. Определитель

$$A = \begin{vmatrix} 1 & 0 & 0.....0 \\ S_2 & S_1 & 1.....0 \\ S_4 & S_3 & S_2.....0 \\ ..... \\ S_{2t-2} & S_{2t-3} & S_{2t-4}...S_{t-1} \end{vmatrix}$$

не равен нулю, если значения S_j, $j = 1, 2, \ldots 2t$, образованы степенными суммами t или $t - 1$ различных элементов поля, и равен нулю, если значения S_j образованы степенными суммами $t - 2$ или меньше различных элементов поля. Кроме того, требуется проверка определителя

$$d = \begin{vmatrix} 1 & 1 & 1.....1 \\ S_1 & 1 & 0.....0 \\ S_3 & S_2 & 1.....0 \\ S_5 & S_4 & S_3.....0 \\ ..... \\ S_{2t-1} & S_{2t-2} & S_{2t-3}..S_{t-1} \end{vmatrix}.$$

При $t = 3$ имеем

$$d = \begin{vmatrix} 1 & 1 & 1 & 1 \\ S_1 & 1 & 0 & 0 \\ S_3 & S_2 & S_1 & 1 \\ S_5 & S_4 & S_3 & S_2 \end{vmatrix} =$$

$$= S_1^3(1 + S_1 + S_1^3) + S_3(1 + S_1 + S_1^2 + S_1^3 + S_3) + S_5(1 + S_1) = 0.$$

Следует отметить, что для поля характеристики $2S_2^2 = S_1^4$, $S_4 = S_1^4$, $S_6 = S_3^2$ и т.д. Для $t = 2$ или $t = 3$ определитель

$$A = \begin{vmatrix} 1 & 0 & 0 \\ S_2 & S_1 & 1 \\ S_4 & S_3 & S_2 \end{vmatrix} = S_1^3 + S_3$$

не равен нулю. Однако $A = 0$ при $t = 1$, поскольку $S_3 = S_1^3$. Рассмотрим пример. С целью упрощения изложения допустим, что $t = 3$. Пусть $X_1 = a^{61}$, $X_2 = a^{60}$ и $X_3 = a^{59}$. Процедура нахождения корней заключает-

ся в том, что в каждом такте значения S_1, S_3 и S_5 умножают на a^1, a^3 и a^5 соответственно. В результате шести таких трансформаций имеем:

	S_1	S_3	S_5	d
начальные значения	a^{22}	a^1	a^4	$\neq 0$
после первого такта	a^{23}	a^4	a^9	$\neq 0$
после второго такта	a^{24}	a^7	a^{14}	$= 0$
после третьего такта	a^{25}	a^{10}	a^{19}	$= 0$
после четвертого такта	a^{26}	a^{13}	a^{24}	$= 0$
после пятого такта	a^{27}	a^{16}	a^{29}	$\neq 0$
после шестого такта	a^{28}	a^{19}	a^{34}	$\neq 0$.

Блок-схема координатного процессора приведена на рис.21. В зависимости от значения t процессор вычисляет значения координат таблич-

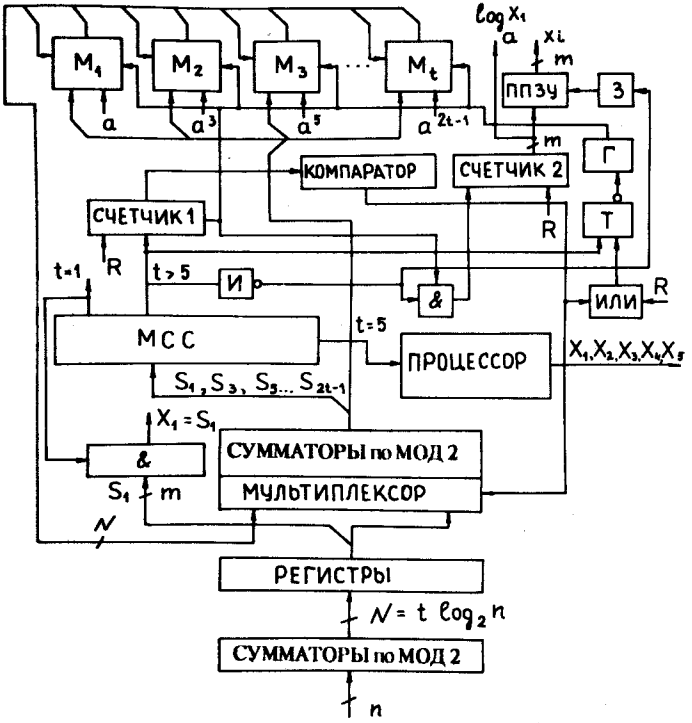


Рис.21. Блок-схема гибридного координатного процессора. Г — генератор импульсов; Т — триггер; З — задержка; И — инвертор; & — элемент И; M_1+M_t — устройства умножения в поле Галуа; R — установка на «0»

ным способом ($t < 5$) или циклическим методом. При этом, если $t = h$ и $h > 5$, количество циклов будет только $h - 5$, после чего процессор снова переходит к решению табличным способом для вычисления остальных пяти корней. Рассмотрим кратко работу процессора. Если мажоритарная схема совпадений вырабатывает сигнал $t > 5$, то происходит установка счетчика 1, и генератор Г начинает вырабатывать тактовые импульсы. С помощью счетчика 2 вычисляется значение степени X_i. В каждом такте значение синдрома S_j умножается на соответствующие константы, и содержимое схем умножения $M_1 + M_t$ складывается с содержимым регистров. Как только в счетчике 1 появляется 5, срабатывает компаратор и начинается процесс вычисления координат табличным методом.

ПРИМЕНЕНИЕ КОДОВ ФАЙЕРА

Нередко при считывании с координатных детекторов последовательным способом регистрируется пакет импульсов, и требуется определить координату центра кластера (рис.22). Сигналы от плоскостей считываются с помощью линий задержки [48]. Экономичная схема кодирования считываемых сигналов на основе кодов Файера предложена в [49]. Коды Файера специально разработаны для исправления пакетов ошибок в технике связи и в вычислительной технике [1]. Таблицы порождающих полиномов для построения кодов Файера длиной $n = 15 \div 1200$ приведены в [50]. Количество разрядов в кодирующем устройстве равно степени r порождающего полинома $g(X)$. Для кластера длиной $b = 3$ и числа сдвиговых импульсов $n = 15$ можно рекомендовать следующий порождающий полином:

$$g(X) = X^9 + X^6 + X^5 + X^4 + X + 1.$$

При этом коэффициент сжатия равен отношению n/r. Причем эффект сжатия растет с увеличением n, поскольку размеры кластера ограничены.

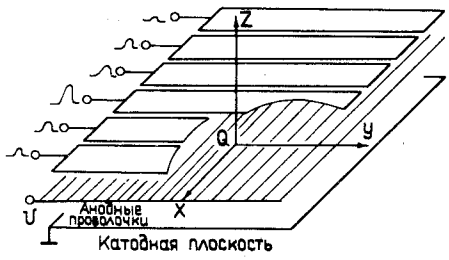


Рис.22. Схема двухкоординатного детектора с катодным считыванием

В [51] показано, что существуют оптимальные коды Файера для $b = 3$ и 4. Например, как это следует из [51], при $b = 3$ вместо 9-разрядного регистра можно применить 6-разрядный регистр и $K_c = 4096/14$. В результате для декодирования такого кода можно использовать ППЗУ.

ПАРАЛЛЕЛЬНЫЕ ШИФРАТОРЫ ДЛЯ ДВУХКООРДИНАТНЫХ ДЕТЕКТОРОВ

В настоящее время разработаны полупроводниковые двухкоординатные (пиксель) детекторы, содержащие множество ячеек с усилителями-формирователями и простейшими схемами для определения границ множественности на основе компараторов [52]. В некоторых разработках применены приоритетные шифраторы, недостатки которых были отмечены выше [53]. В данном разделе будет показано, каким образом метод синдромного кодирования может быть использован для шифрации координат и определения множественности событий, регистрируемых в двухкоординатных детекторах. Кроме того, имеется возможность также эффективно решить проблему распознавания «духов», так как использование приоритетных шифраторов даже при $t = 2$ не решает проблемы в том случае, когда они используются для чтения данных отдельно по координатам X и Y. Если для этих целей применить метод синдромного кодирования, то возможны два подхода. 1) Применение итеративных кодов (см. ниже). 2) При использовании второго метода предполагается применение алгебраической теории кодирования БЧХ-кодов [54].

Предположим, что детектор состоит из $n=2^m-1=k\times k$ ячеек, расположенных в двумерной плоскости X, Y, и $m>2$. В качестве примера на рис.23 приведена блок-схема детектора, содержащего 49 ячеек ($k=7$). С целью упрощения принимаем, что $t = 2$, и сработали ячейки с координатами

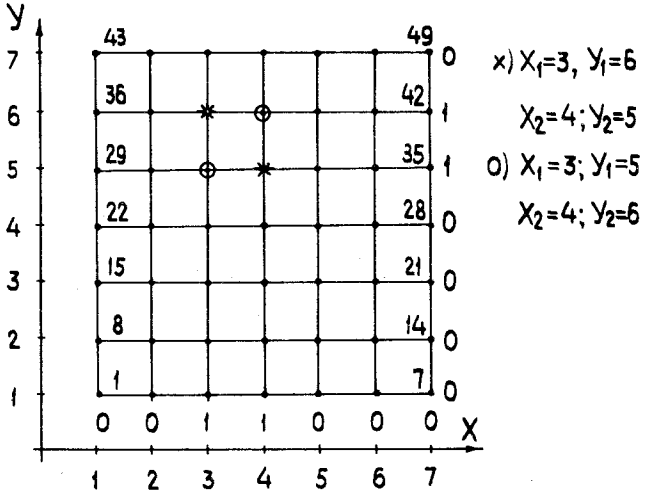


Рис.23. Схематическое изображение двухкоординатного детектора, содержащего 49 пикселей; * и o — события

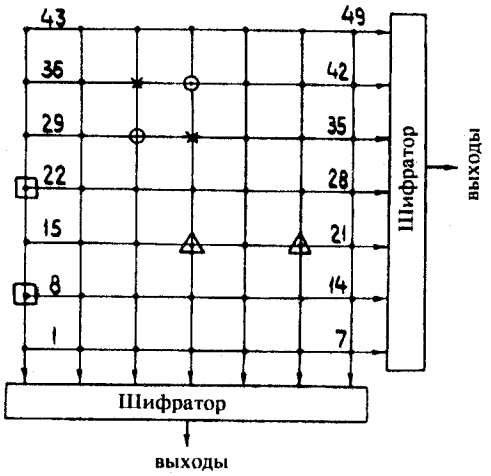


Рис.24. Схематическое изображение детектора с параллельными шифраторами для каждого столбца и каждой колонки; *, o, □ и Δ — события

$X_1=3$, $Y_1=6$ и $X_2=4$, $Y_2=5$ одновременно. В другом случае одновременно сработали ячейки с координатами $X_1=3$ и $Y_1=5$, $X_2=4$, $Y_2=6$. Если данные считываются на регистры, расположенные по координатам X и Y (на рис.23 не показаны), то возникают неопределенности в вычислении координат событий, даже если применить приоритетные регистры. Для разрешения данной проблемы рассмотрим два случая: 1) использование отдельных параллельных шифраторов для каждой строки и каждого столбца отдельно; 2) применение одного параллельного шифратора для всех каналов считывания n.

Так, для $t=2$ и $m=3$ имеем следующую кодирующую матрицу $H_{7,2}$:

Номер ячейки в строке или в столбце						
1		a^0	a^0		100	100
2		a^1	a^3		010	110
3		a^2	a^6		001	101
4	$H_{7,2}=$	a^3	a^2	$\longrightarrow$	110	001
5		a^4	a^5		011	111
6		a^5	a^1		111	010
7		a^6	a^4		101	011
					$\downarrow$ S_1	$\downarrow$ S_3

На рис.24 приведена блок-схема детектора, в котором используются раздельные параллельные шифраторы по строкам и столбцам. Как видно из табл.9, при таком подходе проблема «духов» решается быстро и просто.

На рис.25 приведена принципиальная схема шифратора для $n=7$. При втором методе кодирования все ячейки нумеруются подряд и используется один параллельный шифратор на все каналы регистрации n. Для

Таблица 9. Кодирование координат сработавших ячеек

	Координата X							Координата Y						
	*	*	o	o	□	□	Δ	*	*	o	o	□	Δ	Δ
$S_1 \longrightarrow$	a^2	a^3	a^2	a^3	a^0	a^0	a^2	a^5	a^4	a^4	a^5	a^0	a^2	a^2
$S_3 \longrightarrow$	a^6	a^2	a^6	a^2	a^0	a^0	a^4	a^1	a^5	a^5	a^1	a^5	a^6	a^6

нашего примера требуется шифратор на 49 входов (рис.26), поэтому выбираем следующие параметры: $m = 6$, $n = 2^6 - 1 = 63$ и $t = 2$. Поскольку значения S_1 и S_3 суть элементы поля $GF(2^6)$, то они представляются в виде

$$S_1 = S_{10}a^0 + S_{11}a^1 + S_{12}a^2 + S_{13}a^3 + S_{14}a^4 + S_{15}a^5,$$
$$S_3 = S_{30}a^0 + S_{31}a^1 + S_{32}a^2 + S_{33}a^3 + S_{34}a^4 + S_{35}a^5. \qquad (26)$$

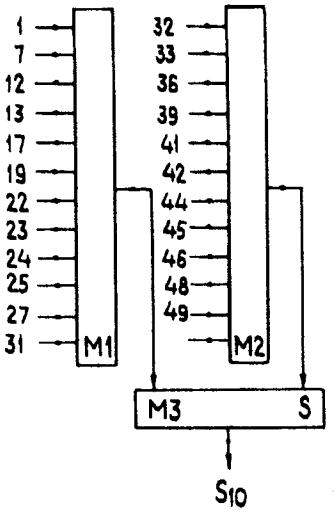


Рис.25. Схема параллельного шифратора для одного ряда (колонки), $t = 2$ и $n = 7$

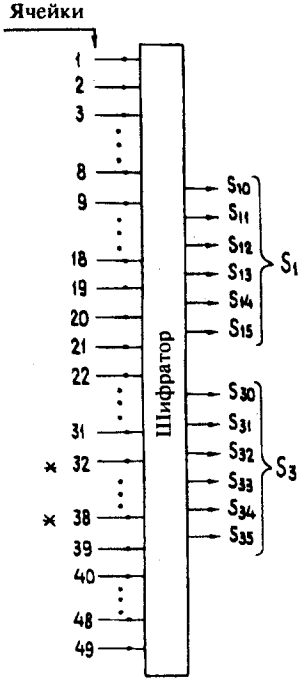


Рис.26. Блок-схема параллельного шифратора для $t = 2$ и $n = 49$

Ниже приводится кодирующая матрица $H_{49,2}$, состоящая из двух колонок, поскольку $t = 2$. Для событий, обозначенных *, имеем

$$\begin{array}{r} 101001 \\ + \\ 001101 \\ \hline \end{array} \qquad \begin{array}{r} 110011 \\ + \\ 101100 \\ \hline \end{array}$$

$$S_1 = 100100 = a^{32} \qquad S_1 = 011111 = a^{57}.$$

Таблица 10. Кодирующая матрица $H_{50,2}$

Номер ячейки	$GF(2^6)$-элементы	Кубы этих элементов	Номер ячейки	$GF(2^6)$-элементы	Кубы этих элементов
1	100000	100000	26	010001	101000
2	010000	000100	27	101000	000101
3	001000	110000	28	011100	111100
4	000100	000110	29	001110	110111
5	000010	101000	30	000111	100010
6	000001	000101	31	110011	011100
7	110000	111100	* 32	101001	110011
8	011000	110111	33	100100	010010
9	001100	100010	34	010010	011010
10	000110	011100	35	001001	011011
11	000011	110011	36	110100	010111
12	110001	010110	37	011010	100110
13	101000	011010	* 38	001101	101100
14	010100	011011	39	110110	110101
15	001010	010111	40	011011	111010
16	000101	100110	41	111101	011111
17	110010	101100	42	101110	100111
18	011001	110101	43	010111	100000
19	111100	111010	44	111011	000100
20	011110	011111	45	101101	110000
21	001111	100111	46	100110	000011
22	110111	100000	47	010011	101000
23	101011	000100	48	111001	000101
24	100101	110000	49	101100	111100
25	100010	000011			

Для оценки определителя $\det L_3$ вычислим значение S_5:

$$S_5 = (a^{31})^5 + (a^{37})^5 = a^{2\times 63}a^{29} +$$

$$+ a^{2\times 63}a^{59} = a^{29} + a^{59} = a^{12}.$$

На рис.27 приведена принципиальная схема для вычисления одной компоненты S_{10} вектора S_1. Если сравнивать два подхода к шифрации, то первый метод целесообразно использовать, когда число ячеек в строках (столбцах) велико.

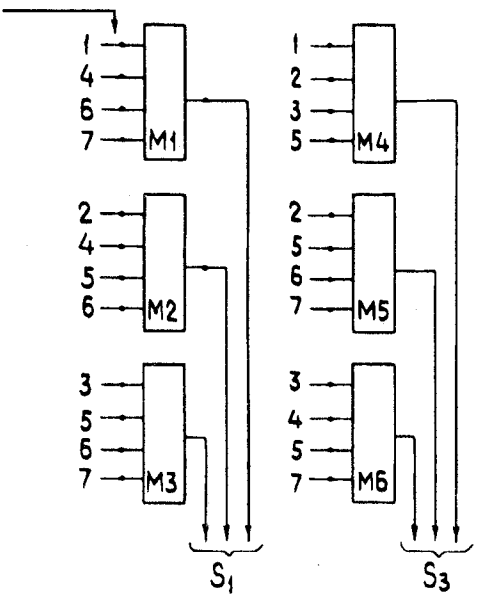


Рис.27. Принципиальная схема вычисления компоненты S_{10} вектора S_1

Обработка сигналов. На конкретном примере рассмотрим другой подход к обработке кода синдрома, который основан на представлении элементов поля в виде полиномов. Для нашего примера имеем

$$\det L_1 = S_1 = a^{32} = 100100 \neq 0,$$

$$\det L_2 = (a^{32})^3 + a^{57} = a^{63}a^{33} + a^{57} = a^0a^{33} + a^{57} \neq 0,$$

$$\det L_3 = a^{192} + a^{96}a^{57} + a^{114} + a^{32}a^{12} = a^{189}a^3 + a^{63}a^{27} + a^{51} + a^{32}a^{12} =$$

$$= a^3 + a^{27} + a^{51} + a^{44} = 0.$$

Нетрудно подсчитать, что, как и следовало ожидать, значение $\det L_3$ равно нулю:

$$\begin{array}{l} \quad 000100 \\ + \\ \quad 011100 \\ + \\ \quad 110101 \\ + \\ \quad 101101 \\ \hline \quad 000000 = \det L_3. \end{array}$$

Представим значения $\det L_1$ и $\det L_2$ в форме

$$\det L_1 = S_1 = a^0 S_{10} + a^1 S_{11} + a^2 S_{12} + a^3 S_{13} + a^4 S_{14} + a^5 S_{15},$$

$$\det L_2 = (S_1^3 + S_3) = a^0 S_{10} + a^1 S_{11} + a^2 S_{12} + a^3 S_{13} + a^4 S_{14} + a^5 S_{15})^3 + \quad (27)$$

$$+ (a^0 S_{30} + a^1 S_{31} + a^2 S_{32} + a^3 S_{33} + a^4 S_{34} + a^5 S_{35}).$$

После возведения в степень и приведения подобных членов имеем следующие значения коэффициентов для члена S_1^3:

$$A_{10} = S_{10} + S_{12} + S_{14} + S_{10}S_{13} + S_{13}S_{15} + S_{11}S_{14} + S_{13}S_{14} + S_{11}S_{15} + S_{12}S_{15},$$

$$A_{11} = S_{12} + S_{10}S_{11} + S_{10}S_{13} + S_{11}S_{13} + S_{12}S_{13} + S_{13}S_{14} + S_{14}S_{15} + S_{11}S_{14},$$

$$A_{12} = S_{14} + S_{10}S_{12} + S_{11}S_{13} + S_{12}S_{14} + S_{12}S_{15} + S_{14}S_{15} +$$

$$+ S_{10}S_{11} + S_{10}S_{14} + S_{11}S_{15},$$

$$A_{13} = S_{11} + S_{13} + S_{15} + S_{11}S_{14} + S_{12}S_{13} + S_{12}S_{14} + S_{12}S_{15} + \quad (28)$$

$$+ S_{13}S_{15} + S_{14}S_{15} + S_{10}S_{13} + S_{10}S_{14},$$

$$A_{14} = S_{13} + S_{14}S_{15} + S_{10}S_{12} + S_{10}S_{14} + S_{10}S_{15} + S_{11}S_{12} +$$

$$+ S_{11}S_{14} + S_{12}S_{14} + S_{12}S_{15} + S_{13}S_{14},$$

$$A_{15} = S_{15} + S_{11}S_{12} + S_{11}S_{13} + S_{11}S_{15} + S_{12}S_{13} + S_{13}S_{15}.$$

Наконец, получаем следующие булевы выражения для $\det L_2$:

$$\det L_2 = \det L_{20} + \det L_{21} + \det L_{22} + \det L_{23} + \det L_{24} + \det L_{25}.$$

$$\det L_{20} = A_{10} + S_{30}$$

$$\det L_{21} = A_{11} + S_{31}$$

$$\det L_{22} = A_{12} + S_{32} \quad (29)$$

$$\det L_{23} = A_{13} + S_{33}$$

$$\det L_{24} = A_{14} + S_{34}$$

$$\det L_{25} = A_{15} + S_{35}.$$

На рис.28 приведена принципиальная схема процессора для определения координат X_1 и X_2 и множественности $t \le 2$. Для преобразования элементов поля S_1 и S_3 в натуральный двоичный код можно применить ППЗУ или ПЛА. С целью получения максимального быстродействия для построения МСС достаточно использовать матрицы совпадений и схемы проверки на четность. Такие устройства могут быть выполнены в виде больших интегральных микросхем. Все логические связи в процессоре описываются с помощью логических уравнений, и поэтому процесс разработки при больших величинах n и t можно автоматизировать.

Быстродействие МСС $T_м$ можно вычислить из соотношения

$$T_м = 2T_ч + 3T_и,$$

где $T_ч$ — задержка в схеме проверки на четность и $T_и$ — задержка в логическом элементе И. Если использовать быстродействующие схемы, то при достаточно больших n и t величина $T_м$ не превышает 10 нс. Таким образом, для построения МСС по методу синдромного кодирования необходимо выполнить следующие процедуры.

1. В соответствии с числом входов n выбирается неприводимый полином m-й степени [1] и вычисляются $2^m - 1$ ненулевых элементов поля $GF(2^m)$. Для $m = 4 \div 10$ можно рекомендовать следующие полиномы: X^4+X+1, X^5+X^2+X+1, X^6+X+1, X^7+X^3+1, $X^8+X^4+X^3+X^2+1$, X^9+X^4+1 и $X^{10}+X^3+1$.

2. Строится матрица проверочных соотношений $H_{n,t}$ для заданного $t \le n/2$. Число колонок в матрице должно быть t.

3. В матрице $H_{n,t}$ элементы поля заменяются на их двоичные эквиваленты.

4. Вычисляются определители t-го порядка.

5. Если для вычисления определителя не используются ППЗУ или ПЛМ, все значения элементов поля представляются в виде полиномов степени $m - 1$ и выполняются необходимые вычисления с целью получения булевых выражений в базисе И и ИСКЛЮЧАЮЩЕЕ ИЛИ.

6. Создается принципиальная схема МСС в соответствии с выражениями (29). Здесь лучше всего применить полупроводниковую технологию. Для вычисления координат событий можно применить гибридный процессор.

Продолжим рассмотрение нашего примера. При $t = 1$ $S_1 = a^{31} = 101001$, т.е. $S_{10} = S_{12} + S_{15} = 1$ и $S_{11} = S_{13} = S_{14} \neq 0$. Тогда $S_3 =$

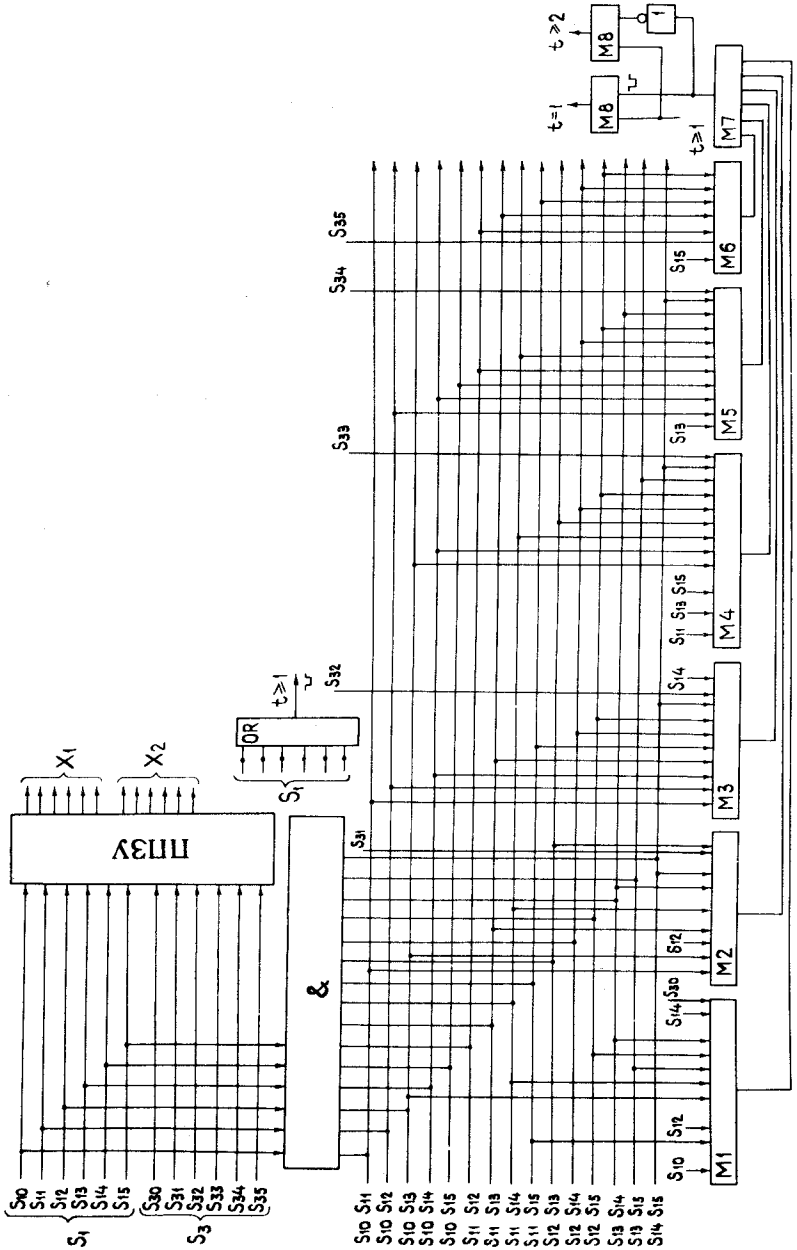


Рис.28. Принципиальная мажоритарная схема совпадений при $T \leq 2$, М1+М7 — МС10160; М8 — МС10102

$= a^{30} = 110011$, т.е. $S_{30} = S_{31} = S_{34} = S_{35} = 1$ и $S_{32} = S_{33} = 0$. Можно проверить, что $\det L_1 = S_1 \neq 0$, но $\det L_2 = 0$. В этом случае логический элемент И (рис.28) открыт, и на выходе МСС формируются сигналы $t > 1$ и $t = 1$. Можно проверить, что сигналы $t \geq 1$ и $t \leq 2$ формируются при $t = 2$.

ПРИМЕНЕНИЕ ТЕОРИИ КОДОВ РИДА — СОЛОМОНА

В ряде экспериментов требуется не только регистрировать координаты отдельных сработавших позиционно-чувствительных детекторов, но и как можно быстрее определить количество кластеров, их координаты и даже образы. Например, по величине кластера, зарегистрированного в калориметре, можно определить энергию взаимодействующих частиц. В координатных детекторах регистрация центра кластера позволяет повысить точность определения координат событий. В таких случаях можно использовать теорию кодов, исправляющих пакеты ошибок. Известно, что в теории кодирования пачка ошибок длиной β определяется вектором ошибки, в котором все единицы заключены в последовательности символов при условии, что крайние символы этой последовательности есть единицы. Так, пачки длиной 6 могут выглядеть следующим образом:

0011111110000 001000010000 001100110000 и т.д.
Пачка 1 Пачка 2 Пачка 3

В работе [55] автор предложил использовать для регистрации таких событий теорию кодов Рида— Соломона (РС-кодов), исправляющих пакеты ошибок [56]. Если положить, что длина кластера $\beta \leq m$, где m — степень неприводимого полинома, то коэффициент сжатия равен $(2^m - 1)/2t$, где $2^m - 1$ — число информационных и $2t$ — число проверочных символов РС-кода, поскольку они имеют длину, состоящую из $n = 2^m - 1$ символов, причем каждый символ содержит m бит. Среди этих символов имеется $2^m - 1 - 2t$ информационных и $2t$ проверочных символов, где t — количество ошибочных символов, исправляемых данным кодом. В общем виде проверочная матрица РС-кода, которая в нашем случае является одновременно и кодирующей матрицей, имеет вид

$$H = \begin{vmatrix} a^0 & a^0 & a^0.. & ....a^0 \\ a^1 & (a^1)^2 & (a^1)^3.. & ...(a^1)^t \\ a^2 & (a^2)^2 & (a^2)^3.. & ...(a^2)^t \\ . & . & . & . \\ . & . & . & . \\ a^{n-1} & (a^{n-1})^2 & (a^{n-1})^3 & (a^{n-1})^t \end{vmatrix}. \qquad (30)$$

При $t = 2$ и $m = 4$ имеем:

$$H_{60,16} = \begin{array}{l|} \text{Группы входов} \\ * \ 0 \\ 1 \\ * \ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \\ 13 \\ 14 \end{array} \begin{vmatrix} a^0 & a^0 & a^0 & a^0 \\ a^1 & a^2 & a^3 & a^4 \\ a^2 & a^4 & a^6 & a^8 \\ a^3 & a^6 & a^9 & a^{12} \\ a^4 & a^8 & a^{12} & a^1 \\ a^5 & a^{10} & a^0 & a^5 \\ a^6 & a^{12} & a^3 & a^9 \\ a^7 & a^{14} & a^6 & a^{13} \\ a^8 & a^1 & a^9 & a^2 \\ a^9 & a^3 & a^{12} & a^6 \\ a^{10} & a^5 & a^0 & a^{10} \\ a^{11} & a^7 & a^3 & a^{14} \\ a^{12} & a^9 & a^6 & a^3 \\ a^{13} & a^{11} & a^9 & a^7 \\ a^{14} & a^{13} & a^{12} & a^{11} \end{vmatrix}. \qquad (31)$$

В матрице (31) значения $a^0 \div a^{14}$ — элементы поля Галуа $GF(2^4)$.

Для определенности положим, что многоканальный детектор имеет 60 каналов регистрации, которые разделены на 15 групп по 4 канала в каждой группе. В свою очередь, группы каналов регистрации пронумерованы степенями элементов поля Галуа $GF(2^4)$, а номера групп каналов, в которых возник кластер, обозначены символом *. Как это принято в алгебраической теории кодирования, полагаем, что ненулевая компонента вектора координат сработавших позиционно-чувствительных детекторов $e(X)$ задается парой элементов Y_i — образом символа кластера и X_i — координатой кластера. Если было t событий, то вектор $e(X)$ имеет t ненуле-

вых компонент, и поэтому для описания таких событий требуется t пар X_i, Y_i. Тогда имеем

$$S_j = \sum_{i=1}^{t} Y_i X_i^j, \quad 1 \le j \le 2t. \tag{32}$$

Для решения уравнений (32) можно использовать рассмотренные выше методы. Для определения количества зарегистрированных кластеров можно воспользоваться теоремой [1]. Матрица

$$L_t = \begin{vmatrix} S_1 & S_2 & S_3 \ldots\ldots S_t \\ S_2 & S_3 & S_4 \ldots\ldots S_{t+1} \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ S_t & S_{t+1} & S_{t+2} \ldots S_{2t+1} \end{vmatrix}$$

невырождена, если S_j образуются точно из t различных ненулевых пар (X_i, Y_i), и матрица (L_t) вырождена, если S_j образуются из меньше чем t ненулевых пар (X_i, Y_i). Другими словами, свойства определителя $\det L_t$ таковы, что если, например, $t = 1$, то $\det L_1 \neq 0$, но зато все остальные определители более высокого порядка равны нулю. Если же $t = 2$, то $\det L_1 \neq 0$, $\det L_2 \neq 0$, но $\det L_3 = 0$ и т.д. При $t = 3$ имеем

$$L_3 = \begin{vmatrix} S_1 & S_2 & S_3 \\ S_2 & S_3 & S_4 \\ S_3 & S_4 & S_5 \end{vmatrix} \quad \begin{array}{l} \det L_1 = S_1 \\ \det L_2 = S_1 S_3 + S_2^2 \\ \det L_3 = S_1 S_3 S_5 + S_1 S_4^2 + S_2^2 S_5 + S_3^3. \end{array} \tag{33}$$

Продолжаем рассмотрение нашего примера. Допустим, что в многоканальном детекторе были зарегистрированы $t = 2$ кластера на позициях $X_1 = a^0$ и $X_2 = a^2$, причем $Y_1 = a^7 = 1101$, $Y_2 = a^{11} = 0111$. Рассмотрим два случая.

а) Кластер $a^{11} = 0111$ поступил на позицию $X_1 = a^2$. Тогда из (32) имеем

$$S_1 = a^{11}a^2 = a^{13}, \; S_2 = a^{11}a^4 = a^{15} = a^0, \; S_3 = a^{11}a^6 = a^2, \; S_4 = a^{11}a^8 = a^4.$$

Кроме того, из соотношений (33) имеем

$$\det L_1 = a^{13} \neq 0, \; \det L_2 = a^{13}a^2 + a^0 = 0, \; \det L_3 = a^{13}a^2a^6 + a^{13}a^8 + a^0a^6 + a^6 = 0.$$

б) Допустим, $X_1 = a^0$, $Y_1 = a^7$ и $X_2 = a^2$, $Y_2 = a^{11}$, т.е. $t = 2$. Далее имеем

$$S_1 = a^7a^0 + a^{11}a^4 = a^5,\ S_2 = a^7a^0 + a^{11}a^4 = a^9,\ S_3 = a^7a^0 + a^{11}a^6 = a^{12}$$

$$\text{и } S_4 = a^7a^0 + a^{11}a^8 = a^3.$$

Для нахождения значений X_1 и X_2 необходимо решить квадратное уравнение

$$X^2 + \sigma_1 X + \sigma_2 = 0. \tag{34}$$

Для частного случая, когда $t = 2$, имеют место соотношения

$$\sigma_1 = \frac{S_3S_2^2 + S_1S_2S_4}{S_2^3 + S_1S_2S_3},\ \ \sigma_2 = \frac{S_2S_4 + S_3^2}{S_2^2 + S_1S_3}, \tag{35}$$

а из равенства (32) получаем связь между S_j, X_i и Y_i:

$$S_1 = X_1Y_1 + X_2Y_2 \text{ и } S_3 = X_1^3Y_1 + X_2^3Y_2. \tag{36}$$

Из равенств (35) имеем $\sigma_1 = a^8$ и $\sigma_2 = a^2$. Можно проверить, что при этих значениях σ_1, σ_2 и $X_1 = a^0$, $X_2 = a^2$ уравнение (34) обращается в тождество. Аналогично в тождество обращаются уравнения (36) при заданных значениях S_1, S_3 и X_1, X_3. Таким образом, по коду синдрома S_j можно с помощью специализированного процессора быстро вычислить множественность t, координаты кластеров X_i и их образы Y_i. Блок-схема процессора описана в [55].

ПРИМЕНЕНИЕ ИТЕРАТИВНЫХ КОДОВ

В ряде случаев вместо кодов с алгебраическими методами декодирования имеет смысл использовать так называемые итеративные коды [57,58]. Такие коды удобно использовать для кодирования информации, регистрируемой в двухкоординатных детекторах. В качестве исходных кодов для построения итеративных кодов можно брать любые известные корректирующие коды. Метод построения двумерного итеративного кода рассмотрим на следующем примере. Во-первых, информационные разряды $X_{ij} = 0, 1$ представляются в виде матрицы:

Информационные символы ↓ Проверки по строкам ↓

$$\begin{array}{|cccc|c|}
X_{11} & X_{12} & X_{13}\ldots\ldots X_{ij}\ldots X_{1m} & & \sum_{j=1}^{m} X_{1j} \\
X_{21} & X_{22} & X_{23}\ldots\ldots X_{2j}\ldots X_{2m} & & \sum_{j=1}^{m} X_{2j} \\
\ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots \\
X_{\alpha 1} & X_{\alpha 2} & X_{\alpha 3}\ldots\ldots X_{\alpha j}\ldots X_{\alpha m} & & \sum_{j=1}^{m} X_{\alpha m} \\
\hline
\sum_{i=1}^{\alpha} X_{i1} & \sum_{i=1}^{\alpha} X_{i2} & \sum_{i=}^{\alpha} X_{i3}\ldots\sum_{i=1}^{\alpha} X_{ij}\ldots\sum_{i=1}^{\alpha} X_{im} & &
\end{array} \qquad (37)$$

Проверки по столбцам

После этого добавляются проверки по строкам и столбцам. На рис.29 приведена блок-схема итеративного кода. Существенным является то, что общее кодовое расстояние d γ-мерного итеративного кода равно $d = d_1 d_2 \ldots d_\gamma$, где γ равно числу кодов, используемых для итерации. Если напомнить известное соотношение $t = (d - 1)/2$, то с помощью такого кода при больших числах n имеется возможность экономичным способом распознавать сложные события с большой множественностью. Наиболее простым итеративным кодом является код типа ИЛИ-ИЛИ и «чет-чет» ($d = 2\times2 = 4$). Как показал анализ [57], код ИЛИ-ИЛИ нашел применение в сцинтилляционных годоскопах с целью экономии ФЭУ при регистрации одночастичных событий с кластерами. Проверочные соотношения кода типа «чет-чет» вычисляются с помощью сумматоров по модулю два, и поэтому ФЭУ для этих целей не подходят, поскольку эти приборы выполняют функции усилителей-смесителей сигналов. Однако итеративный код «чет-чет» можно использовать эффективно, если кодируемые сигналы имеют логические уровни. Рассмотрим два рисунка. На рис.30 показаны картины событий в двумерной плоскости при t=1÷3. Причем позиции событий безотносительны. Будем подсчитывать количество событий путем счета количества признаков «чет» или «нечет»

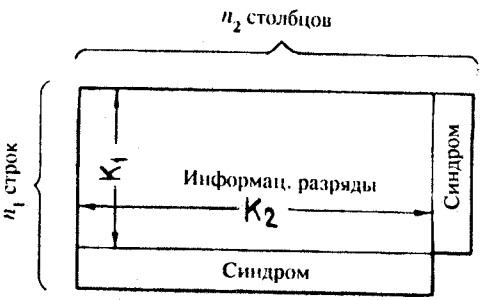


Рис.29. Структура двумерного итеративного кода

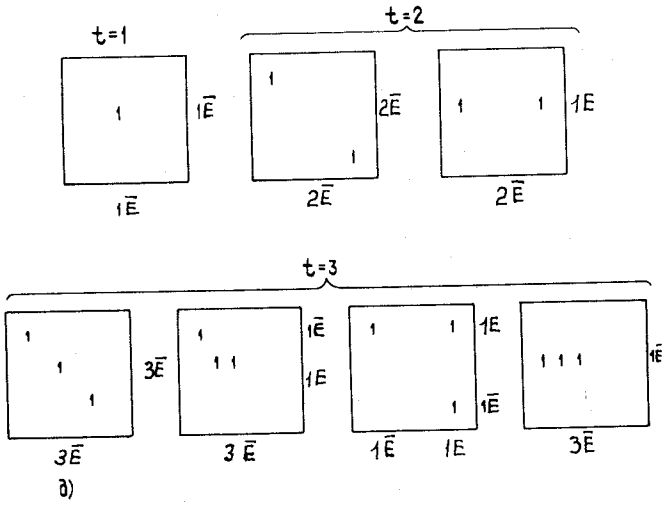


Рис.30. Вид событий в двумерной плоскости; $t = 1{+}3$. E — «чет»; $\overline{E}$ — «нечет»

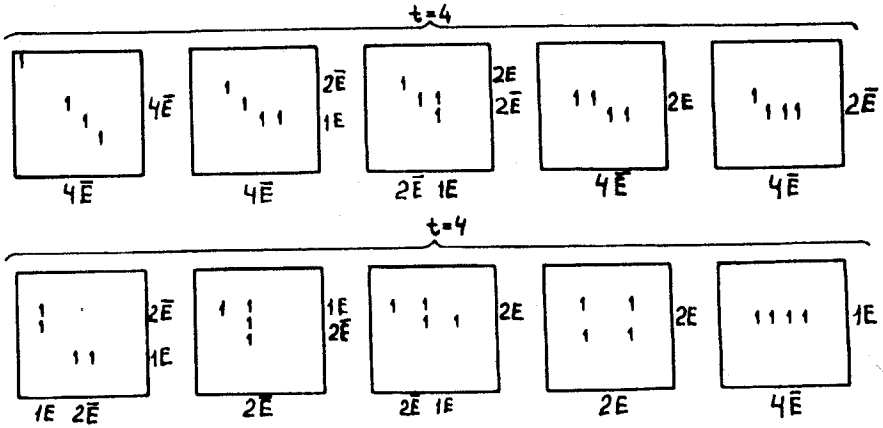


Рис.31. Значение четностей при $t = 4$

отдельно по координатам X и Y. Как видно из рис.31, при $t = 1$ получается один «нечет» по X-координате и один «нечет» по координате Y. При $t = 2$, 3 и 4 и имеются 2, 4 и 10 различных картинок соответственно. На рис.31 приведены аналогичные картинки для $t = 4$ (E — «чет» и $\overline{E}$ — «нечет»). Причем нет необходимости приводить симметричные картинки, дающие

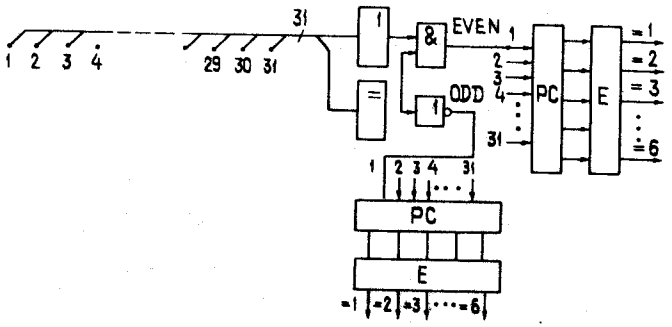


Рис.32. Часть схемы параллельного счетчика для двухкоординатного детектора, содержащего 961 ячейку. РС — параллельный счетчик; 1 — ИЛИ; & — И; Е — дешифратор

одинаковый результат. Таким образом, если с помощью параллельных счетчиков подсчитывать количество сигналов «чет» и «нечет», то можно построить экономичное и быстродействующее устройство [59] для регистрации множественности в одно- и двухкоординатных детекторах. Например, предположим, что двухкоординатный детектор содержит $31 \times 31 = 961$ ячейку (31 строка и 31 столбец). С целью упрощения на рис.32 приведена основная часть схемы для одной строки (столбца). Поскольку нуль — число четное, то требуется некоторое количество логических элементов ИЛИ для того, чтобы не считывать нули. Быстродействие такого счетчика T_c можно вычислить из соотношения

$$T_c = T_ч + 2T_и + T_{сч} + T_д,$$

где $T_ч$ — задержка в схеме проверки на четность, $T_и$ — задержка в логическом элементе И, $T_{сч}$ — быстродействие (31,5)-счетчика и $T_д$ — задержка в дешифраторе. Если использовать ЕСЛ-логику, то величина T_c не превышает 43 нс! [60]. Естественно, что для регистрации больших значений t необходимо выбирать более мощные коды (коды с большими d). Возможны два подхода к выбору кодов с большими d. 1) К простейшему двумерному итеративному коду добавить еще две диагональные проверки. 2) Применить коды с большими кодовыми расстояниями. Известно, что модифицированный код Хэмминга имеет кодовое расстояние $d = 4$. Тогда двумерный итеративный код Хэмминг — Хэмминг имеет $d = 16$ и $t = 15$. В дополнение к этому, как следует из теории, с помощью такого кода можно одновременно регистрировать $\nu = t/2$ координат.

Подведем некоторые итоги. Если сравнивать эффективность кодов с алгебраической структурой, например БЧХ-кодов и итеративных кодов, то можно отметить следующее. Коды с алгебраической структурой относительно просто декодируются. Кроме того, поскольку теоретической базой таких кодов является теория конечного поля Галуа, то наряду с процессами кодирования и декодирования над полученными данными можно легко реализовать различного рода логические и алгебраические операции. Синдром итеративного кода представляет собой хаотичный набор двоичных символов, и поэтому для его декодирования лучше всего подходят табличные методы на основе ППЗУ и ПЛМ.

ПРИМЕНЕНИЕ СУПЕРПОЗИЦИОННЫХ КОДОВ

В рассматриваемых примерах в основном для вычисления синдрома использовались сумматоры по модулю два, для построения которых необходимы инверторы, а входные и выходные сигналы должны быть логическими. Существует также интересный, как с теоретической, так и, особенно, с практической точки зрения, класс так называемых суперпозиционных кодов [61]. Отличительной особенностью таких кодов является то, что в качестве схем для вычисления синдрома можно использовать обычные усилители-смесители сигналов, в том числе и ФЭУ. Другими словами, кодируемые сигналы могут быть как электрическими, произвольной формы и амплитуды, так и световыми. В частности, автором было показано, что такие популярные коды, как обыкновенный код Хэмминга и код Грэя, являются суперпозиционными кодами [64].

Ниже будет показано, каким образом разработанный автором ряд таких кодов и кодирующих устройств может быть использован для пост-

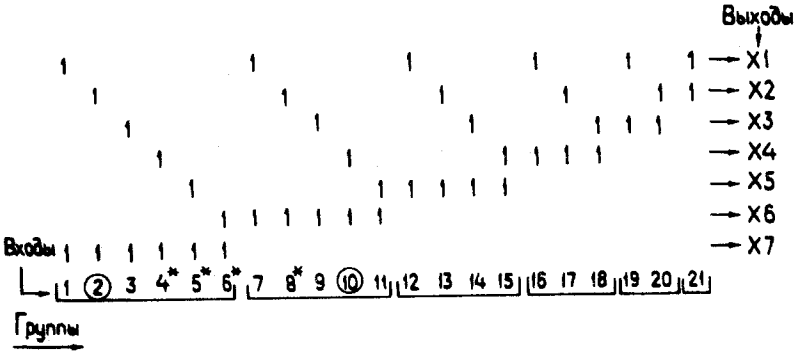


Рис.33. Кодирующая матрица $H_{28,7}$

роения эффективных схем отбора событий. Работа одного из таких устройств основана на свойстве кодирующей матрицы C_N^2 (число сочетаний из N по два) [62,63]. Матрица содержит $n = C_N^2$ столбцов и N строк. Причем в каждом столбце содержится только по две единицы (коэффициент разветвления сигнала равен двум), а коэффициент сжатия $K_c = C_N^2/N$. Например, пусть $n = 28$, тогда $N = 7$. На рис.33 приведена схема кодирующей матрицы $H_{28,7}$, а на рис.34 показано, каким образом вычисляются 1, 3, 4 и 8-й разряды синдрома. В качестве разветвителей сигналов могут быть использованы гибкие световоды. Используя такую схему кодирования, можно при оптимальном значении коэффициента сжатия однозначно регистрировать одночастичные события с тройными кластерами, что нетрудно проверить, складывая по правилам булевой суммы два и три соседних столбца [64]. Кроме того, такую кодирующую схему можно выполнить в виде маски, как это показано на рис.35. Рассмотрим два случая [64]. 1) В годоскопической плоскости зарегистрирована одна частица без кластера, т.е. $t = 1$. Тогда вес синдрома равен двум. Поскольку все столбцы кодирующей матрицы различны, то 7-разрядный код синдрома несет в себе информацию как о том, что $t = 1$, так и о координате сработавшего позиционно-чувствительного детектора, которая

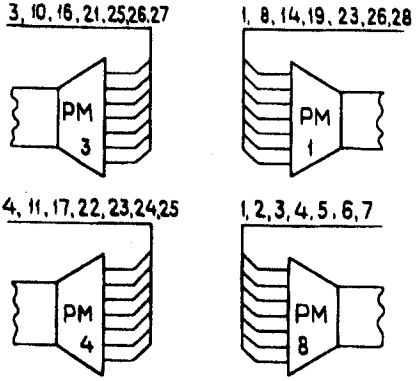


Рис.34. Схема для вычисления 1, 3, 4 и 8-го разрядов синдрома. РМ — ФЭУ

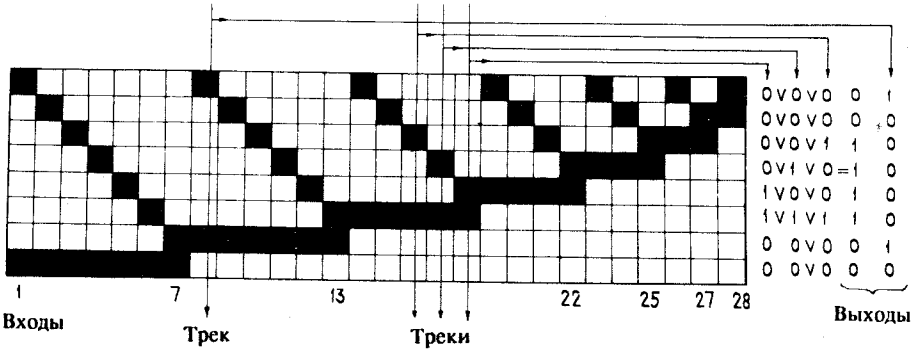


Рис.35. Маска детектора для $t = 1$; $n = 28$ и $N = 7$

декодируется просто с помощью ППЗУ. 2) Если же был зарегистрирован двойной или тройной кластер, то вес синдрома равен трем или четырем, а значения кодов различаются при всех комбинациях одного, двух и трех сработавших соседних датчиков.

Счетчик кластеров. Кодирующая матрица $H_{64,8}$, состоящая из восьми повторяющихся единичных матриц восьмого порядка I_8, может быть использована для построения экономичного счетчика кластера с $1 \le b \le 8$.

$$H_{64,8} = \left|\begin{array}{c} 10000000 \\ 01000000 \\ 00100000 \\ 00010000 \\ 00000100 \\ 00000010 \\ 00000010 \\ 00000001 \end{array}\right| \left|\begin{array}{c} \\ \\ \\ \\ I_8 \\ \\ \\ \end{array}\right| \left|\begin{array}{c} \\ \\ \\ \\ \cdots\cdots \\ \\ \\ \end{array}\right| \left|\begin{array}{c} \\ \\ \\ \\ I_8 \\ \\ \\ \end{array}\right| \left|\begin{array}{c} N \\ \downarrow \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array}\right|$$

$$n \longrightarrow \quad 1 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 57 \ldots\ldots 64$$

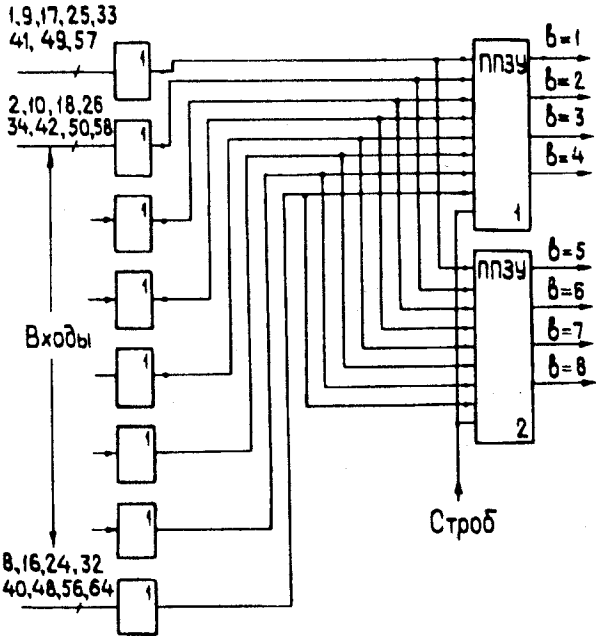


Рис.36. Схема счетчика кластеров для $n = 64$; $N = 8$ и $b = 8$

На рис.36 приведена блок-схема счетчика. Модуль ППЗУ программируется таким образом, что если вес синдрома $w = 1$, то $b = 1$, если же вес $w = 2$, то $b = 2$ и т.д. В общем виде для построения счетчика при $b = z$ нужно использовать единичную матрицу порядка z.

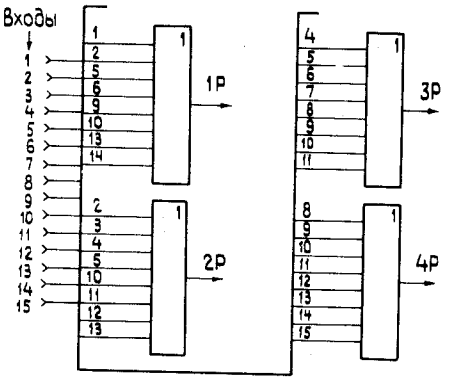


Рис.37. Схема вычисления синдрома для одной строки: 1 — ИЛИ; 1P+4P — выходы

Суперпозиционные итеративные коды. С целью увеличения кодового расстояния суперпозиционных кодов, можно для итерации использовать популярный код Хэмминга ($d = 3$) и код Грэя ($3 \le d \le 4$) в сочетании с простой схемой кодирования, когда по одной из координат проверки вычисляются с помощью логических элементов ИЛИ [57]. Допустим, что двухкоординатный детектор содержит 225 ячеек, организованных в виде квадратной матрицы, содержащей $k = 15$ строк и 15 столбцов. Например, по строкам будем вычислять проверки в соответствии с (15,4)-кодом Грэя, а по столбцам используем обычные элементы ИЛИ ($d = 2$). В результате получаем следующую кодирующую матрицу для одной строки, столбцы которой, по существу, представляют собой 4-разрядные слова $H_{15,4}$ кода Грэя:

$$H_{15,4} = \begin{array}{c|c|c} & & N \\ & & \downarrow \\ & 110011001100110 & 1 \\ & 011110000111100 & 2 \\ & 000111111110000 & 3 \\ & 000000011111111 & 4 \\ n & 1.....15 & \end{array}$$

На рис.37 приведена принципиальная схема вычисления синдрома для одной строки. В целом для данной схемы требуется 15 элементов ИЛИ на 15 входов и 4×15 элементов ИЛИ на 8 входов. Кодовое расстояние такого итеративного суперпозиционного кода равно 8. Следует отметить, что на практике имеет смысл вначале провести моделирование событий для каждого конкретного эксперимента и детектора, а затем можно уже разработать для быстрой и оптимальной кодировки соответствующий код.

ВЫВОДЫ

В современных и вновь планируемых экспериментах по физике высоких и сверхвысоких энергий наметилась тенденция к выполнению противоречивых с точки зрения электроники и вычислительной техники, условий: достижение высокого быстродействия отбора полезных событий в условиях регистрации большого количества информации и поиска редких событий на уровне большого фона. Стало очевидным, что решение столь непростой задачи требует нетрадиционного подхода к созданию быстродействующих и многоуровневых систем. Суть метода синдромного кодирования заключается в том, что многие задачи отбора полезных событий, которые традиционно решались методами обычной двоичной арифметики, в которой ряд операций выполнить довольно сложно, можно решать с помощью алгебры конечных полей. Причем, прежде чем анализировать события, данные, считываемые от многоканальных детекторов заряженных частиц, предварительно сжимаются в отношении $n/\log_2 n$.

Кроме того, показано, что, используя методы вычислений в поле Галуа $GF(2^m)$ с применением аналитических вычислений на ЭВМ, можно выполнять синтез довольно сложных и в том числе программно-управляемых логических модулей, параллельных шифраторов, мажоритарных схем совпадений и специализированных процессоров для быстрого отбора физических событий. Важно также, что реализация универсального динамически программируемого модуля позволит в будущем создавать сложные триггерные системы, не содержащие механических контактов, которые можно будет быстро перепрограммировать на решение различных физических задач.

ПРИЛОЖЕНИЕ

ЭЛЕМЕНТЫ ПОЛЯ ГАЛУА ПО МОДУЛЮ $X^6 + X + 1$

$a^0 = 100000$	$a^{21} = 110111$	$a^{42} = 010111$
$a^1 = 010000$	$a^{22} = 101011$	$a^{43} = 111011$
$a^2 = 001000$	$a^{23} = 100101$	$a^{44} = 101101$
$a^3 = 000100$	$a^{24} = 100010$	$a^{45} = 100110$
$a^4 = 000010$	$a^{25} = 010001$	$a^{46} = 010011$

$a^5 = 000001$	$a^{26} = 111000$	$a^{47} = 111001$
$a^6 = 110000$	$a^{27} = 011100$	$a^{48} = 101100$
$a^7 = 011000$	$a^{28} = 001110$	$a^{49} = 010110$
$a^8 = 001100$	$a^{29} = 000111$	$a^{50} = 001011$
$a^9 = 000110$	$a^{30} = 110011$	$a^{51} = 110101$
$a^{10} = 000011$	$a^{31} = 101001$	$a^{52} = 101010$
$a^{11} = 110001$	$a^{32} = 100100$	$a^{53} = 010101$
$a^{12} = 101000$	$a^{33} = 010010$	$a^{54} = 111010$
$a^{13} = 010100$	$a^{34} = 001001$	$a^{55} = 011101$
$a^{14} = 001010$	$a^{35} = 110100$	$a^{56} = 111110$
$a^{15} = 000101$	$a^{36} = 011010$	$a^{57} = 011111$
$a^{16} = 110010$	$a^{37} = 001101$	$a^{58} = 111111$
$a^{17} = 011001$	$a^{38} = 110110$	$a^{59} = 101111$
$a^{18} = 111100$	$a^{39} = 011011$	$a^{60} = 100111$
$a^{19} = 011110$	$a^{40} = 111101$	$a^{61} = 100011$
$a^{20} = 001111$	$a^{41} = 101110$	$a^{62} = 100001$
		$a^{63} = a^0 = 100000$

СПИСОК ЛИТЕРАТУРЫ

1. **Питерсон У.** — Коды, исправляющие ошибки: Пер. с англ. М.: Мир, 1964.
2. **Ancheta T.C.** — IEEE Trans. on Inform. Theory. 1976, vol.IT—22, p.432—436.
3. **Frohwerk R.A.** — Hewlett Packard Journal, 1977, vol.28, No.9. p.2—8.
4. **Nikityuk N.M., Radzhabov R.S., Shafranov M.D.** — Nucl. Instr. and Meth. 1977, vol.155, p.485—489.
5. **Никитюк Н.М., Раджабов Р.С., Шафранов М.Д.** — ПТЭ, 1978, № 4, с.95—98.
6. **Никитюк Н.М.**— Препринт ОИЯИ, Р11-84-484, Дубна, 1984.
7. **Гайдамака З.И., Калинников В.А., Никитюк Н.М. и др.** — Препринт ОИЯИ, Р13-82-628, Дубна, 1982.
8. **Никитюк Н.М.** — Препринт ОИЯИ, Р10-87-254, Дубна, 1987.
9. **Nikityuk N.M.** — JINR Preprint E10-90-184, Dubna, 1990. In: Proc. of the AAECC-8 Conf. Tokio, 1990, Lecture Notes in Computer Science, Springer Verlag, 1990, в.508, p.144—154.
10. **Gustafson L., Hagberg E.A.** — Nucl. Inst. and Meth., 1988, vol.A265, p.521—532.
11. **Reed I.S., Truong T.K.** — IEEE Trans. on Inf. Theory, 1977, vol.IT-21, p.657—661.
12. **Both T.L.** — IRE Trans. on Circuit Theory, 1975, vol.CT-10, p.279—281.
13. **Menger K.C.** — IEEE Trans. on Computers, 1969, vol.C-18, p.241—250.

14. Берлекэмп Е.Р. — Алгебраическая теория кодирования. М.: Мир, 1970.
15. Kain R.Y. — IEEE Trans. on Computers, 1970, vol.C-19, p.249—254.
16. English W.R. — IEEE Trans. on Computers, 1981, vol.C-30, p.225—229.
17. Bartee C.T., Schneider P.I. — IRE Trans. on Inf. Theory, 1962, vol.IT-8, p.17—24.
18. Benjauthrit B., Reed I. — IEEE Trans. on Computers, 1976, vol.C-25, p.78—86.
19. Bartee T.C., Schneider P.I. — Information and Control, 1963, vol.6, p.79—98.
20. Berlecamp E.R., Rumsey H., Solomon G. — Information and Control, 1967, vol.10, p.553—564.
21. Александров И.Н., Гайдамака Р.И., Никитюк Н.М. и др. — Препринт ОИЯИ, Р10-84-865, Дубна, 1984.
22. Gaidamaka R.I., Nikityuk N.M. — JINR Preprint, E10-88-53, Dubna, 1988.
23. Nikityuk N.M. — JINR Preprint, E11-87-10, Dubna, 1987.
24. Blauhut R.E. — Proc. IEEE. 1985, vol.73, p.30—53.
25. Benjauthrit B, Reed I. — IEEE Trans. on Computers, 1978, vol.C-27, p.757—763.
26. Pradhan D.K. — IEEE Trans. on Computers, 1978, vol.C-27, p.239—248.
27. Wesselkamper T.C. — IEEE Trans. on Computers, 1978, vol.C-27, p.232—238.
28. Александров И.Н., Гайдамака Р.И., Никитюк Н.М. и др. — Препринт ОИЯИ, Р10-84-865, Дубна, 1984.
29. Никитюк Н.М. — Препринт ОИЯИ, Р11-85-365, Дубна, 1985.
30. Никитюк Н.М. — Авт. свид. СССР № 1236457. G06 7/00. ОИ, 1986, № 21, с.199.
31. Никитюк Н.М. — Препринт ОИЯИ, Р11-87-54, Дубна, 1987.
32. Никитюк Н.М. — Препринт ОИЯИ, Р11-88-852, Дубна, 1988.
33. Jonson M., Lankford A.J., Amendolia S. et al. — Preprint SLAC-PUB-4611, Stanford University, 1988.
34. Nikityuk N.M. — In: Proc. the I Int. Joint Conf. of ISSAC-88 and AAECC-6, Roma, 1988, Springer-Verlag, Ed. T. Mora. 1988, vol.357, p.324—335; JINR Preprint, E10-88-28, Dubna, 1988.
35. Никитюк Н.М. — Препринт ОИЯИ, Р10-87-254, Дубна, 1987.
36. Nikityuk N.M. — JINR Preprint, E-89-362, Dubna, 1989.
37. Messy J.L. — IEEE Trans. on Inf. Theory, 1965, vol.IT-11, p.580—585.
38. Banerji R.B. — Proc. IRE, 1961, vol.49, p.1585.
39. Блох Э.Л. — Известия АН СССР. Техническая кибернетика. 1964, № 3, с.30-37.
40. Polkinhorn F. — IEEE Trans. on Inf. Theory, 1966, vol.IT-12., p.480—481.
41. Chien R.T., Cunningham D.B., Oldham I.B. — IEEE Trans. on Inf. Theory, 1969, vol.IT-15, p.329—335.
42. Okano H., Imai H. — IEEE Trans. on Computers, 1987, vol.C-36, p.1165—1171.
43. Никитюк Н.М. — Препринт ОИЯИ Р10-88-853, Дубна, 1988.
44. Никитюк Н.М. — Препринт ОИЯИ Р10-89-16, Дубна, 1989.
45. Berlecamp E.R. — IEEE Trans. on Inf. Theory, 1965, vol.IT-11, p.577—579.
46. Hurst S.L. — IEEE Trans. on Computers, 1981, vol.C-30, p.986—987.
47. Chien R.T. — IEEE Trans. on Inf. Theory, 1964, vol.IT-10, p.357—362.
48. Jeavons A.P., Fora N., Lindberg D. et al. — IEEE Trans. on Nucl. Sciense, 1976, vol.NS-23, p.250—260.
49. Никитюк Н.М. — Препринт ОИЯИ, Р10-88-742, Дубна, 1988.
50. Wagner W. — IEEE Trans. on Inf. Theory, 1970, vol.IT-16, p.649—650.
51. Elspas B. — IRE Trans. on Inf. Theory, 1962, vol.IT-8, p.30—42.
52. Dierichx B. — Nucl. Instr. and Meth., 1989, vol.A275, p.542—544.
53. Parker S. — Nucl. Instr. and. Meth., 1989, vol.A275, p.494—516.
54. Nikityuk N.M. — JINR Preprint, E10-91-161, Dubna, 1991.
55. Никитюк Н.М. — Препринт ОИЯИ, Р10-88-854, Дубна, 1988.
56. Shao H.M., Truong T.K., Leslie J. et al. — IEEE Trans. on Computers. 1985, vol.C-34, p.393—402.

57. **Никитюк Н.М.** — Препринт ОИЯИ, Р10-87-266, Дубна, 1987.
58. **Calingaert P.** — Journal of the Association for Computing Machinary, 1961, vol.8, p.186—200.
59. **Никитюк Н.М.** — Авт. свид. СССР, № 15446992. ОИ, 1990, № 8, с.245.
60. **Nikityuk N.M.** — JINR Preprint, E10-91-567, Dubna, 1991.
61. **Kautz W.H., Singleton R.C.** — IEEE Trans. on Inf. Theory, 1964, vol.IT-10, p.363—377.
62. **Никитюк Н.М.** — ПТЭ, 1986, № 3, с.59—65.
63. **Никитюк Н.М., Рукояткин Р.А., Светов А.Л.** — ПТЭ, 1991, № 1, с.95—97.
64. **Никитюк Н.М.** — ПТЭ, 1983, No.3, с.74—81.
65. **Хетагуров Я.А., Руднев Ю.П.** — Повышение надежности цифровых устройств методами избыточного кодирования. М.: Энергия, 1974.