ЛЕКЦИИ
ДЛЯ МОЛОДЫХ
УЧЕНЫХ

Н.М.Никитюк

# От современной алгебры - к специализированным процессорам

ДУБНА

*ЛЕКЦИИ ДЛЯ МОЛОДЫХ УЧЕНЫХ*

*Выпуск 40*

*РЕДАКЦИОННЫЙ СОВЕТ*



...ий Дубна, 1987

**ОБЪЕДИНЕННЫЙ ИНСТИТУТ ЯДЕРНЫХ ИССЛЕДОВАНИЙ**

Р10-87-401

Н.М.Никитюк



# ОТ СОВРЕМЕННОЙ АЛГЕБРЫ - К СПЕЦИАЛИЗИРОВАННЫМ ПРОЦЕССОРАМ





Дубна 1987

## I. Введение

В последнее десятилетие интенсивно развивается такое важное направление в информатике, как цифровая обработка сигналов, которая в теоретическом аспекте базируется на методах абстрактной алгебры, таких, как группы, поля, кольца быстрые алгоритмы и проч. Как это отмечено в работе /I/, без знания основных положений современной алгебры трудно понять состояние теоретических и прикладных исследований в области цифровой обработки сигналов и эффективное использование БИС и микропроцессоров.

В течение длительного времени цифровая обработка сигналов, задачи которой обычно формулируются в системах действительных и комплексных чисел, и задачи кодирования с обнаружением и с исправлением ошибок формулируются в другой числовой системе, называемой полем Галуа и обозначаемой $GF(p)$. На самом деле с методической точки зрения весьма важно изложить эти дисциплины с единых позиций, так как в работах /I,2/ показано, что многие известные методы цифровой обработки сигналов /например, дискретное преобразование Фурье/ вполне применимы в области кодирования с обнаружением и исправлением ошибок и, наоборот, алгебраическая теория кодирования вполне применима в области цифровой обработки сигналов.

Теория кодирования является прикладной наукой, и поэтому она черпает свои задачи из техники связи, радиолокации, измерительной и особенно вычислительной техники. Однако наиболее широкое применение получили лишь самые простые способы кодирования, такие, как проверка на четность, коды Хэмминга, циклические коды с обнаружением ошибок. Практическое применение таких кодов в настоящее время весьма широко, так как выпускаются специальные микросхемы для вычисления проверок на четность. Достаточно подробно и популярно простейшие коды описаны в работе /3/, а способы применения некоторых из них в ядерной электронике приведены в статье /4/.

Однако, как это отмечают специалисты по теории кодирования, более мощные корректирующие коды не нашли широкого применения из-за сложности восприятия инженерами математического аппарата теории. Такое положение вряд ли можно считать нормальным, так как методы современной алгебры в сочетании с аналитическими вычислениями на ЭВМ открывают большие перспективы их применения для расчета переключательных функций, создания универсальных динамически программируемых логических модулей и специализированных процессоров /5,6 /. Более того, в последние годы интенсивно развивается новое научно-техническое направление , которое получило название сигнатурного анализа /7,8/ или синдромного тестирования /9/. Суть его в том,

что с появлением микропроцессоров и приборов со встроенными БИС и микропроцессорами возникает необходимость контроля и обнаружения неисправностей в компонентах без их удаления из устройства. Возникла необходимость в создании простых и эффективных тестовых приборов, обеспечивающих отыскание неисправного компонента испытываемой аппаратуры. Простейшие тестовые приборы - сигнатурные анализаторы - в качестве основных логических узлов содержат сдвиговые регистры с логической обратной связью, которые описываются с помощью полиномов. Другими словами, инженерам необходимы знания основ современной алгебры.

## 2. Поле Галуа

Полем называется множество, в котором однозначно определен результат сложения и умножения любых двух элементов. Следует отметить, что понятие элемент поля не совпадает в принципе с термином логический элемент, который широко известен из вычислительной техники. Поле содержит 0 и I. Сложение и умножение ассоциативны и коммутативны, а умножение, как обычно, дистрибутивно относительно сложения. Другими словами, для любых трех элементов поля $a$, $b$ и $c$ имеем:

$$a(b+c) = ab + ac.$$

Каждый элемент $a$ имеет единственный противоположный элемент $-a$, такой, что $a+(-a)=0$. Каждый ненулевой элемент $a$ имеет единственный обратный элемент $1/a$, такой, что $a(1/a) = I$. Для каждого элемента $a$ выполняются равенства:

$$0+a = a = 1\cdot a \quad \text{и} \quad 0\cdot a = 0.$$

Порядком поля называется число его элементов. Если порядок поля бесконечен, то оно нызывается бесконечным полем. Примером бесконечного поля является множество рациональных чисел. Если же порядок поля конечен, то оно называется конечным.

Определение: поле, содержащее конечное число элементов $p$, называется конечным полем и обозначается $GF(p)$ ($GF$ означает *Galois Field* - поле Галуа).

Так как каждое поле должно содержать нулевой и единичный элементы, то оно должно содержать, как минимум, два элемента. Дополненные таблицами сложения и умножения

| + | 0 | I |
|---|---|---|
| 0 | 0 | I |
| I | I | 0 |

| · | 0 | I |
|---|---|---|
| 0 | 0 | 0 |
| I | 0 | I |

два таких элемента и образуют поле, а именно поле $GF(2)$. Других полей, содержащих два элемента, не существует. Поле $GF(2)$ называется еще основным полем.



Складывая последовательно единицы конечного поля, получим неограниченный ряд элементов

$$I,\ I+I,\ I+I+I,\ \ldots$$

Так как число элементов поля конечно, то в указанном ряду найдется сумма $p$ единиц, равная нулевому элементу поля. В работе [12] показано, что наименьшее число, удовлетворяющее этому условию, является простым числом. Простое число $p$, обладающее свойством $p = 0$ в конечном поле, называется характеристикой этого поля. Например, в поле $GF(2)$ число $p = 2$, так как $I+I = 0 \mod 2$.

Расширенное поле Галуа. Основное поле Галуа можно расширить и образовать поле большего размера. Упорядоченный способ получения расширенного поля основан на использовании некоторой конструкции, которую можно назвать полиномиальным представлением расширенного поля. Один из наиболее распространенных способов построения расширенного поля Галуа заключается в следующем. В конечном поле существуют неприводимые полиномы определенных степеней. Напомним, что неприводимые полиномы обладают тем свойством, что их невозможно разложить на два полинома меньшей степени. Пусть $P(X)$ есть неприводимый полином степени $m$ с коэффициентами из поля $GF(2)$ и пусть $a$ является корнем полинома $P(X)$. Тогда расширенное поле создается путем добавления элемента $a$ к полю $GF(2)$. Тогда порядок или число элементов расширенного поля равен $2^m$, и такое поле обозначается $GF(2^m)$. Например, поле $GF(2^8)$ можно построить как расширение поля $GF(2)$, используя полином

$$P(X) = X^8 + X^4 + X^3 + X^2 + 1.$$

Этот полином имеет коэффициенты в $GF(2)$ и неразложим на множители в $GF(2)$. Следовательно, поле $GF(2^8)$ состоит из всех 8-разрядных двоичных чисел, рассматриваемых как полиномы степени не более семи. Сложение является покомпонентным сложением в поле $GF(2)$, иными словами, поразрядной операцией "Исключающее ИЛИ". Умножение состоит из умножений полиномов, приводимых по модулю $P(X)$. Следует отметить, что построение расширенного поля при $m = 8$ вручную представляет значительные трудности. Для этих целей можно использовать специально созданные программы для вычислений на ЭВМ [13]. Другие примеры расширенных полей приведены в следующем параграфе.

Модульные операции. Над конечными полями справедливы почти все основные результаты классической алгебры. Сюда относится алгебра многочленов, теория определителей и метод решения систем уравнений, теория матриц и проч. Однако результаты вычислений принимают конечную форму. Этот факт не умаляет большого прикладного значения модулярной алгебры в современной вычислительной технике.

## 3. Примеры расширенных полей

Прежде чем рассматривать примеры построения расширенных полей Галуа и их применение для построения специализированных процессоров и в других областях техники, например, в приборостроении, ниже приводится еще одно, но более формальное определение расширенного поля Галуа.

В работе /II/ доказывается следующая важная теорема. Пусть $p(X)$ - многочлен с коэффициентами из поля $F$. Алгебра многочленов над полем F по модулю $p(X)$ является полем тогда и только тогда, когда многочлен $p(X)$ - неприводим в поле F, т. е. если $p(X)$ нельзя представить в виде произведения многочленов с коэффициентами из F. На основе этой теоремы расширенным полем степени $m$ над F называется поле, образованное многочленами над полем F по модулю неприводимого многочлена $p(X)$ степени $m$ над F. Ниже приводятся примеры построения расширенных полей.

Рассмотрим три неприводимых полинома $X^2 + X + I$, $X^3 + X + I$, и $X^4 + X + I$. Таблицы неприводимых полиномов над полем $GF(2)$ вплоть до 34-й степени приведены в приложении В /II/. Отметим, что знак + обозначает сумму по модулю два.

Одним из простейших расширенных полей является поле $GF(2^2)$, образованное над полиномом $X^2 + X + I$. Такое поле имеет четыре элемента: $a^0 = I0$, $a^1 = 0I$, $a^2 = II$ и нулевой элемент $0 = 00$. Элемент $a^1$, который называется примитивным, является корнем полинома $X^2 + X + I$ и поэтому имеем $a^2 + a + I = 0$, в чем нетрудно убедиться, складывая столбиком двоичные эквиваленты элементов поля. Далее, имеем $a^2 = a + I$. Другими словами, зная корень неприводимого полинома, нетрудно построить нужное поле, учитывая при этом тот факт, что в поле Галуа $GF(2^m)$ имеется $m$ линейно независимых элементов, которые образуют базис данного поля. Так в поле $GF(2^2)$ базис состоит из двух элементов: $a^0 = I0$ и $a = 0I$. Остальные два элемента уже нам известны. Элемент a, являющийся корнем полинома называют еще элементом, образующим поле. Роль и значение этого элемента в образовании поля отмечается в следующей теореме: в конечном поле порядка p существует такой элемент $a$, что каждый ненулевой элемент этого поля может быть представлен как некоторая степень элемента a, т. е. мультипликативная группа конечного поля носит циклический характер. /I2/. Практически это значит, что поскольку $a^2 = a + I$, то, умножив обе части этого равенства на a, получим $a^3 = a^2 + a = a^0$. Аналогично $a^4 = a^3 + a^2 = a^0 + a^1 = a$, $a^5 = a^3 = a^2$ и т. д. И, вообще, конечное поле порядка p содержит примитивный элемент /порядка p - I/, степени которого пробегают все ненулевые элементы поля. Другими словами,



если поле содержит элемент a, то оно должно содержать и все степени $a, a^2, a^3, \ldots a^{2^m-2}$. Далее, так как поле содержит мультипликативный обратный каждого ненулевого элемента, то ему принадлежат также a, $a^{-1}, a^{-2}, \ldots a^{-(2^m-2)}$. Наименьшее из положительных чисел $n$, для которого $a^n = I$ /здесь I - вектор/ т. е. $a^n = a^0$, называется порядком элемента a, или то же самое: порядок элемента равен числу различных степеней этого элемента. Например, порядок элемента a в поле $GF(2^2)$ равен $2^m - I = 2^2 - I = 3$. Порядок нулевого элемента не определен. Суммируя вышеизложенное, можно отметить следующее. Ненулевые элементы расширенного поля можно представить тремя способами в виде: I/ степеней примитивного элемента, 2/ линейной комбинации базисных элементов, 3/ двоичных чисел. В табл. I - 3 даны представления элементов для трех полей.

Таблица I.

Элементы поля $GF(2^2)$ $\quad a^2 + a + I = 0$

| | | |
|---|---|---|
| $0 = 0$ | 00 | $GF(2)$ |
| $a^0 = I$ | I0 | $GF(2)$ |
| $a = a$ | 0I | $GF(2^2)$ |
| $a^2 = I + a$ | II | $GF(2^2)$ |

По аналогии, а также используя рассмотренные правила и теоремы, можно построить еще два поля $GF(2^3)$ и $GF(2^4)$

Таблица 2.

Элементы поля $GF(2^3)$ $\quad (a^3 + a + I = 0)$

| | | |
|---|---|---|
| $0 = 0$ | 000 | $GF(2)$ |
| $a^0 = I$ | I00 | $GF(2)$ |
| $a^1 = a$ | 0I0 | $GF(2^3)$ |
| $a^2 = a^2$ | 00I | $GF(2^3)$ |
| $a^3 = I + a$ | II0 | $GF(2^3)$ |
| $a^4 = a + a^2$ | 0II | $GF(2^3)$ |
| $a^5 = I + a + a^2$ | III | $GF(2^3)$ |
| $a^6 = I + a^2$ | I0I | $GF(2^3)$ |
| $a^7 = I = a$ | I00 | $GF(2^3)$ |

и т. д.

Таблица 3

Элементы поля $GF(2^4)$ $a^4 + a + I$

| | | |
|---|---|---|
| $0 = 0$ | 0000 } $GF(2)$ | |
| $a^0 = I$ | I000 } | |
| $a = a$ | 0I00 | |
| $a^2 = a^2$ | 00I0 | |
| $a^3 = a^3$ | 000I | |
| $a^4 = I + a$ | II00 | |
| $a^5 = a + a^2$ | 0II0 | |
| $a^6 = a^2 + a^3$ | 00II | |
| $a^7 = I + a + a^3$ | II0I | $GF(2^4)$ |
| $a^8 = I + a^2$ | I0I0 | |
| $a^9 = a + a^3$ | 0I0I | |
| $a^{10} = I + a + a^2$ | III0 | |
| $a^{11} = a + a^2 + a^3$ | 0III | |
| $a^{12} = I + a + a^2 + a^3$ | IIII | |
| $a^{13} = I + a^2 + a^3$ | I0II | |
| $a^{14} = I + a^3$ | I00I | |
| $a^{15} = I = a^0$ | I000 | |

и т. д.

Из табл. 2 видно, что в поле $GF(2^3)$ имеется уже три базисных /линейно независимых/элемента : $a^0 = I00$, $a = 0I0$ и $a^2 = 00I$. Подставляя в полином $X^3 + X + I$ элемент a, получим $a^3 = a + I$. После этой операции имеем $a^4 = a^2 + a = 0II$, $a^5 = a^3 + a^2 = a + I + a^2 = III$, $a^6 = a^4 + a^3 = a^2 + a + a + I = a^2 + I = I0I$, $a^7 = a^3 + a = a + I + a = I = I00 = a^0$, $a^8 = a$, так как $a^8 = a^7 a = a^0 a = a$. По аналогии читатель может рассмотреть получение элементов в поле $GF(2^4)$.

Следует подчеркнуть, что в поле Галуа, в отличие от известного бесконечного поля с позиционной системой счисления, нет таких понятий, как "одно число больше другого". Поэтому в представлении элементов в виде двоичных слов не имеет значения, какой разряд является старшим и какой младшим.

С практической точки зрения представляет интерес построение устройств для получения последовательности элементов поля Галуа. Такие устройства называются счетчиками в поле Галуа. Они представляют собой сдвиговые регистры с логической обратной связью. Причем число разрядов регистра и вид обратных связей определяется полиномом, образующим данное поле. Так, на рис. I приведена схема счетчика в поле $GF(2^3)$. Поскольку $a^3 = a + I$, то сигнал переноса должен поступать на входы первого и второго разрядов регистра, а сигнал



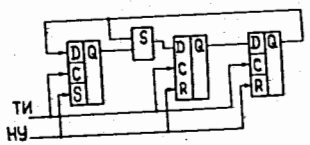



Рис. I. Счетчик в поле Галуа $GF(2^3)$.
НУ- начальная установка.
ТИ-тактовые импульсы.

| + | $a^0$ | $a^1$ | $a^2$ | $a^3$ | $a^4$ | $a^5$ | $a^6$ |
|---|---|---|---|---|---|---|---|
| $a^0$ | 0 | $a^3$ | $a^6$ | $a^1$ | $a^5$ | $a^4$ | $a^2$ |
| $a^1$ | $a^3$ | 0 | $a^4$ | $a^0$ | $a^2$ | $a^6$ | $a^5$ |
| $a^2$ | $a^6$ | $a^4$ | 0 | $a^5$ | $a^3$ | $a^1$ | $a^0$ |
| $a^3$ | $a^1$ | $a^0$ | $a^5$ | 0 | $a^6$ | $a^2$ | $a^4$ |
| $a^4$ | $a^5$ | $a^2$ | $a^1$ | $a^6$ | 0 | $a^0$ | $a^3$ |
| $a^5$ | $a^4$ | $a^6$ | $a^3$ | $a^2$ | $a^0$ | 0 | $a^1$ |
| $a^6$ | $a^2$ | $a^5$ | $a^0$ | $a^4$ | $a^3$ | $a^1$ | 0 |

Рис.2.Таблица сложения двух элементов в поле $GF(2^3)$.

| X | $a^0$ | $a^1$ | $a^2$ | $a^3$ | $a^4$ | $a^5$ | $a^6$ |
|---|---|---|---|---|---|---|---|
| $a^0$ | $a^0$ | $a^1$ | $a^2$ | $a^3$ | $a^4$ | $a^5$ | $a^6$ |
| $a^1$ | $a^1$ | $a^2$ | $a^3$ | $a^4$ | $a^5$ | $a^6$ | $a^0$ |
| $a^2$ | $a^2$ | $a^3$ | $a^4$ | $a^5$ | $a^6$ | $a^0$ | $a^1$ |
| $a^3$ | $a^3$ | $a^4$ | $a^5$ | $a^6$ | $a^0$ | $a^1$ | $a^2$ |
| $a^4$ | $a^4$ | $a^5$ | $a^6$ | $a^0$ | $a^1$ | $a^2$ | $a^3$ |
| $a^5$ | $a^5$ | $a^6$ | $a^0$ | $a^1$ | $a^2$ | $a^3$ | $a^4$ |
| $a^6$ | $a^6$ | $a^0$ | $a^1$ | $a^2$ | $a^3$ | $a^4$ | $a^5$ |

Рис. 3. Таблица умножения двух элементов в поле $GF(2^3)$

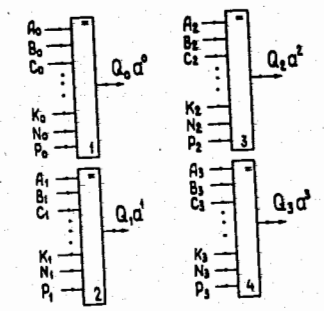



Рис. 4. Принципиальная схема сумматора на I2 входов в поле $GF(2^4)$.

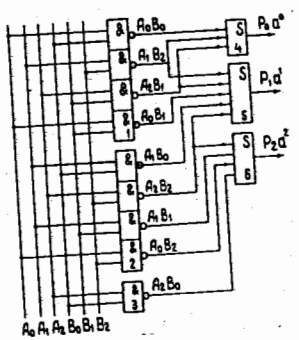



Рис. 5. Принципиальная схема умножения двух элементов в поле $GF(2^3)$.

переноса от первого разряда на второй приходит через сумматор по модулю два, так как $a^3 = I + a$. Если в начальный момент времени в первый разряд /слева/ занести единицу, то в результате последующих сдвигов состояния регистра будут принимать значения, соответствующие степеням элемента a. Фактически мы здесь имеем схему умножения на элемент a. /II с. 204/.

## 4. Основные алгебраические операции над элементами поля Галуа

Как мы уже отмечали выше, элементы поля нередко представляются в виде полиномов степени $m-I$ /см. табл. I - 3/. Так, в поле $GF(2^3)$ любые два элемента A и B можно представить в виде:
$A = A_0a^0 + A_1a + A_2a^2$ и $B = B_0a^0 + B_1a + B_2a^2$. Аналогично в поле $GF(2^4)$ два элемента A и B имеют вид: $A = A_0a^0 + A_1a \ldots + A_3a^3$ и $B = B_0a^0 + B_1a + \ldots + B_3a^3$. Причем коэффициенты $A_0, A_1, \ldots A_3$ и $B_0, B_1, \ldots B_3$ принимают только значения 0 или I. Что касается базисных элементов, то они принимают те значения, которые присущи данному полю.
Так же, как и в обычной двоичной арифметике при выполнении операций над элементами поля, имеются некоторые отличия в правилах выполнения операций "вручную" и машинным способом /I0 - I5 /. Это касается таких операций, как умножение, деление, возведение в степень и извлечение квадратного корня. Операции умножения и деления вручную выполняются очень просто: степень произведения элементов равна сумме степеней сомножителей, причем суммирование выполняется по модулю $2^m - I$.
Операция деления элемента A на элемент B равносильна операции умножения элемента A на обратный элемент $B^{-1}$, который определяется из условия $B \cdot B^{-1} = I$. Так, для элемента $a^6$ в поле $GF(2^4)$, обратным элементом является элемент $a^9$, так как $a^6 a^9 = a^{15} = a^0 = I$. = I000.
На рис. 2 и 3 приведены таблицы сложения и умножения двух элементов поля Галуа $GF(2^3)$. Вручную обратный элемент получается просто: Надо из числа $2^m - 1$ вычесть степень данного элемента и тогда величина разности будет равна степени обратного элемента. Например, обратным элементом для $a^2$ в поле $GF(2^3)$ будет элемент $a^5$, так как $7 - 2 = 5$. Проверка дает: $a^2 a^5 = a^7 = a^0$.
Операция возведения в степень элементов поля выполняется так же, как и возведение обычных чисел, с той лишь разницей, что она выполняется с учетом цикличности группы, т. е. по модулю $2^m - I$. Например, в поле $GF(2^3)$ $(a^4) = a^{12} = a^7 a^5 = a^5$ или $(a^5)^3 = a^{35} = (a^7)^5 = a^0 =$ = I. Операции сложения и вычитания равносильны и выполняются по модулю два. Например, $a^5 + a^5 = a^5 - a^5 = 0$.



С практической точки зрения представляет интерес, какими современными аппаратными средствами реализуются операции над элементами поля Галуа ? Для выполнения операции сложения используются такие микросхемы, как К500ИЕI60, КI55ИП2, К53IИП5 и I55ЛП5, которые представляют собой схемы проверки на четность или сумматоры по модулю два. Поскольку операция суммирования является модульной, то можно в одном такте одновременно складывать большое число элементов. Предварительно суммируются соответствующие коэффициенты, которые используются в представлении элементов поля в виде полиномов. Рассмотрим конкретный пример. Как известно, микросхема типа К500ИЕI60 имеет I2 входов и поэтому , используя такие схемы, можно без дополнительного каскадирования выполнить суммирование I2 элементов поля. На рис. 4 приведена принципиальная схема такого сумматора. Видно, что операция суммирования сводится всего лишь к определению четности в двоичных словах, состоящих из однотипных коэффициентов в виде представлений элементов поля с помощью полиномов. Следует отметить, что вышеперечисленные микросхемы путем каскадирования можно использовать для сложения практически любого числа слагаемых. Далее, если внимательно рассмотреть табл. I - 3, где в третьих колонках приведены двоичные эквиваленты элементов поля, то получается, что число единиц в каждом столбце четно. Это значит, что сумма всех различных элементов поля равна нулю. В остальных случаях в силу цикличности поля при суммировании всякий раз получается один из элементов поля. Например, если сложить I4 элементов поля $GF(2^4)$ , начиная сверху, получим элемент суммы, равный $a^{14}$.
Теперь на конкретном примере рассмотрим, каким образом можно построить схему для параллельного умножения двух элементов поля $GF(2^3)$.
Имеем: $A \cdot B = (A_0a^0 + A_1a + A_2a^2)(B_0a^0 + B_1a + B_2a^2) = A_0B_0a^0a^0 + A_0B_1a^0a^1 + A_0B_2a^0a^2 + A_1B_0a^1a^0 + A_1B_1a^2 + A_1B_2a^3 + A_2B_0a^2a^0 + A_2B_1a^3 + A_2B_2a^4$.
Учитывая, что $a^3 = a + I$ и $a^4 = a + a^2$, получим: $A_0B_0 + A_0B_1a + A_0B_2a^2 + A_1B_0a + A_1B_1a^2 + A_1B_2 + A_1B_2a + A_2B_0a^2 + A_2B_1 + A_2B_1a + A_2B_2a + A_2B_2a^2$ (I).
Выписав отдельно члены выражения (I) с одинаковыми базисными элементами, получим окончательно булевы функции, с помощью которых можно построить принципиальную схему умножения двух элементов поля $GF(2^3)$:

$$\begin{aligned} &A_0B_0 + A_1B_2 + A_2B_1 && \langle a^0 \rangle \\ &A_0B_1 + A_1B_0 + A_1B_2 + A_2B_1 + A_2B_2 && \langle a \rangle \\ &A_0B_2 + A_1B_1 + A_2B_0 + A_2B_2 && \langle a^2 \rangle \end{aligned} \qquad (2)$$

Знаки $\langle a^0 \rangle$, $\langle a \rangle$ и $\langle a^2 \rangle$ означают , что это есть коэффициенты при базисных элементах в выражении для произведения двух элементов A и B .
Таким образом, для аппаратной реализации операции умножения двух

элементов с помощью обычных микросхем необходимо выполнить логическую операцию И над всевозможными парами коэффициентов и затем выполнить операцию сложения в соответствии с вычисленными выражениями типа (2). Далее, если в соотношениях (2) положить A = B ,то получим булевы выражения для возведения элемента $A$ в квадрат:

$$\begin{aligned} &A_0A_0+\cancel{A_1A_2}+\cancel{A_2A_1}= A_0 && a^0\\ &\cancel{A_0A_1}+\cancel{A_1A_0}+\cancel{A_1A_2}+\cancel{A_2A_1}+ A_2A_2= A_2 && a \qquad (3)\\ &\cancel{A_0A_2}+ A_1A_1+\cancel{A_2A_0}+ A_2A_2 = A_1+ A_2 && a^2 . \end{aligned}$$

Следует напомнить, что поскольку операция суммирования выполняется по модулю два, то при приведении подобных членов необходимо руководствоваться следующим правилом: сумма четного числа одинаковых членов равна нулю, а сумма нечетного числа членов равна одному из членов суммы. Кроме того, если в результате каких-либо преобразований появляются члены, степень которых превышает $2^m - 1$, то они разлагаются по базисным элементам.

По аналогии нетрудно найти переключательные функции для возведения элемента A в куб:

$$\left[A_0a^0 + A_2a + (A_1 + A_2)a^2\right]\left(A_0a^0 + A_1a + A_2a^2\right) = A^3$$

или тоже самое:

$$\begin{aligned} &A_0+ A_2+ A_1A_2 +A_1 && \langle a^0\rangle\\ &A_0A_1+ A_1 + A_2A_0 && \langle a\rangle\\ &\quad A_0A_1+ A_2 && \langle a^2\rangle . \end{aligned}$$

Такой итерационный процесс можно продолжить, и в случае необходимости получить логические выражения для возведения элемента A и в другие степени.

Вычисление инверсного элемента мы здесь не рассматриваем. Такой алгоритм детально описан в работах (II,I6) .

Следует отметить, что, в отличие от обычной алгебры, где можно получить общие выражения и закономерности для произвольных чисел, в модулярной алгебре те или иные соотношения действительны для конкретного поля $GF(2^m)$. Так, для того , чтобы получить рассмотренные нами выше соотношения для поля $GF(2^4)$, нужно повторить необходимые вычисления с учетом того, что в этом поле любые два элемента A и B можно представить в виде: $A = A_0a^0 + A_1a + A_2a^2 + A_3a^3$ и $B_0a^0 + B_1a + B_2a^2 + B_3a^3$, где коэффициенты $A_0$, $A_1$, $A_2$, $A_3$, $B_0$, $B_1$, $B_2$ и $B_3$ могут принимать только значения 0 или I. Например, в элементе $a^{12}$ в поле $GF(2^4)$ все коэффициенты равны единице.

На рис. 4 и рис. 5 в качестве примера приведены принципиальные схемы для умножения двух элементов в поле $GF(2^3)$ и сложения I2 элементов в поле $GF(2^4)$.

Следует отметить, что при величине $m \geq 4$ вычисления вручную становятся громоздкими и даже практически невозможными. Поэтому для автоматизации процесса вычислений в поле Галуа созданы программы, в которых используются системы аналитических вычислений на ЭВМ /I3/.

На базе таких вычислений созданы таблицы. Одна из таких таблиц приведена на рис. 6.



## 5. Совмещенные операции в поле Галуа

Рассматривая практическую реализацию выполнения ряда операций в поле Галуа, автор заметил, что имеется возможность просто и одновременно в течение одного такта выполнять такие операции, как умножение нескольких элементов поля или умножение элемента B на куб элемента A и т. д. Практическое применение таких процедур будет рассмотрено в следующем разделе.

Для простоты изложения рассмотрим , каким образом можно получить булевы выражения для умножения элемента B на квадрат элемента A в поле $GF(2^3)$. Имеем: $B = B_0a^0 + B_1a + B_2a^2$ и $A^2 = A_0^2a^0 + A_2a + (A_1 + A_2)a^2$. Умножив B на A, после приведения подобных, получим:

$$\begin{aligned} &B_0A_0+ B_1A_1+ B_1A_2+ B_2A_2 && \langle a^0\rangle\\ &B_0A_2+ B_1A_0+ B_1A_1+ B_1A_2+ B_2A_1+ B_2A_2 && \langle a\rangle\\ &B_0A_1+ B_0A_2+ B_1A_2+ B_2A_0+ B_2A_2 && \langle a^2\rangle . \end{aligned}$$

Аналогичные булевы выражения можно получить для произведений типа $B\cdot A^3$ , $B\cdot A^2\cdot C$ и т.д., где C - третий элемент этого же поля.

На рис. 7 приведена таблица, содержащая результаты такой совмещенной операции, как $B\cdot A^3$. Подобные таблицы используются автором для программирования ППЗУ /см.ниже/. Следует обратить внимание на то, что результат совмещенных операций зависит от перестановки сомножителей. Например, пусть $B = a^4$ и $A = a^6$. Тогда $B\cdot A^3 = a^4(a^6)^3 = a^4a^{18} = a^4a^3 = a^7$. В то же время $A\,B^3 = a^3$. Этот вывод имеет важное практическое значение, так как позволяет избегать ошибки в процессе программирования ППЗУ. Нетрудно было бы показать, что в поле Галуа допустимы также совмещенные операции типа $B/A^2$, $B/A^3$... и т. д.

В следующем разделе мы рассмотрим, каким образом вышеизложенные свойства полей Галуа можно использовать для построения специализированного процессора и других устройств дискретной логики.

## 6. Специализированный процессор в поле Галуа

Рассмотрим конкретный пример. Допустим, что имеется годоскоп, содержащий I5 сцинтилляторов. По условию эксперимента необходимо как можно быстрее выработать сигналы о том, что через сцинтилляторы прошла одна и только одна частица $t = I$, две и только две $t = 2$, три и только три частицы $t = 3$ или $t \geq 4$. Кроме того, требуется оп-

| 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. 6. Таблица умножения двух элементов в поле $GF(2^4)$

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

Рис. 7. Таблица умножения элемента B на элемент $A^3$ в поле $GF(2^4)$.



ределить координаты сцинтилляторов, от которых поступили сигналы, и выработать сигнал при условии срабатывания наперед заданной комбинации сцинтилляторов.

Традиционно такая задача решается следующим образом. С помощью мажоритарной схемы совпадений находятся неравенства типа $\geq 1$, $\geq 2$, $\geq 3$ и т. д. Координаты сработавших сцинтилляторов определяются с помощью приоритетных шифраторов, для чего требуется определенное время для последовательного опроса тех разрядов регистра, в которых записаны единицы.

Используя метод синдромного кодирования, поставленную задачу можно решить в ряде случаев более эффективным способом /I4/. Кратко суть метода заключается в следующем. Вначале рассмотрим, как это принято в практике кодирования, типичную многоканальную систему передачи данных, как это показано на рис. 8,а. На передающей стороне имеется регистр, содержащий $\ell$ разрядов. С помощью кодирующего устройства к $\ell$- разрядному слову по определенному правилу добавляется $K$ разрядов синдрома /контрольные проверки/, так что по каналу передачи к приемному устройству передается $K+\ell = n$ - разрядное слово. На приемной стороне с помощью декодирующего устройства полученное слово анализируется с целью нахождения ошибочных позиций. Если в процессе передачи данных произошло искажение некоторых разрядов, то декодер, используя избыточные /контрольные проверки/ данные, обнаруживает и исправляет ошибки, а информационные символы регистрируются в $\ell$- разрядном регистре.

Теперь рассмотрим более простую схему /рис. 8,б/. Допустим, что передаваемое $n$ - разрядное слово всегда равно нулю, но в процессе передачи такого слова к нему добавляются ошибочные символы. Например, передаем I5-разрядное нулевое слово

000000000000000,

а на приемной стороне получено следующее слово

0I00I0I00000000,

т.е. возникли ошибки во втором, пятом и седьмом разрядах, если счет разрядов вести слева направо. Если число проверочных символов выбрано так, что имеется возможность исправлять не менее трех ошибок, то число разрядов на выходе декодера будет $N = 3\log_2 I5 - I2$. Таким образом, имеется эффект сжатия данных с I5 бит до I2 бит. Причем эффект сжатия данных существенно растет с ростом величины $n$ при условии, что $t \ll n$. Такое условие на практике довольно часто выполняется.

Для построения процессора прежде всего необходимо определить структуру декодера или, как его еще называют в теории кодирования,

"устройство для вычисления синдрома принятого кодового слова".В работе /4/ автором было показано, что для кода Хэмминга, с помощью которого исправляются одиночные ошибки, устройством для вычисления синдрома служит обыкновенный параллельный шифратор,с помощью которого унитарный позиционный код преобразуется в двоичный код, удобный для выполнения над ним различного рода арифметических операций. Однако такой шифратор может правильно функционировать при условии, что величина $t$ =I. Поскольку в практике кодирования существуют декодеры, с помощью которых вычисляется синдром для кода, исправляющего $t>1$ ошибок, то естественно возникла идея использовать такие устройства для одновременной регистрации $t$ событий в годоскопических системах. Особенно перспективным оказалось использование для этих целей так называемых БЧХ-кодов, имеющих алгебраическую структуру /I0, II/. В теории кодирования дается точный ответ на поставленный выше нами вопрос о структуре декодера, которая очень наглядно задается с помощью матрицы проверочных соотношений. На рис. 9 приведена одна из таких матриц , соответствующая БЧХ-коду, исправляющему три ошибки при $n$= I5. Здесь изображены две равносильные матрицы. Первая матрица слева состоит из последовательности элементов поля $GF(2^4)$. Причем в первой колонке содержатся все ненулевые элементы в порядке возрастания их степеней, а вторая колонка состоит из кубов соответствующих им элементов первой колонки. В третьей колонке помещены элементы в пятой степени. Справа на рисунке приведены двоичные эквиваленты элементов, расположенных соответственно слева.

Получилось так, что последний столбец в матрице $H^T$ состоит из нулей, а два соседних столбца - одинаковы. Поэтому синдром в данном конкретном случае представляет собой I0-разрядное слово, для декодирования которого можно использовать быстродействующее запоминающее устройство емкостью IK. Для определенности положим, что сигналы поступили одновременно от 4-го, 9-го и I3-го сцинтилляторов. Эти позиции отмечены символом * . Как видно из рис.9, каждой строке матрицы $H^T$ поставлен в соответствие сцинтиллятор годоскопа. Обозначив координаты сработавших сцинтилляторов через $X_1$, $X_2$, и $X_3$, / в элементах поля Галуа $GF(2^4)$ /, получим по аналогии с теорией кодирования

$$S_1 = X_1 + X_2 + X_3 \ , \ S_3 = X_1^3 + X_2^3 + X_3^3 \text{ и } S_5 = X_1^5 + X_2^5 + X_3^5, \qquad (4)$$

где $S_1 = a^3 + a^8 + a^{12} = a$ , $S_3 = a^9 + a^9 + a^6 = a^6$ и $S_5 = a^0 + a^{10} + a^0 = a^{10}$.
В соответствии с алгебраической теорией кодирования для нахождения координат сработавших сцинтилляторов необходимо подставить всевозможные элементы поля $GF(2^4)$ в следующее уравнение и выделить те элементы, при которых это уравнение обращается в тождество:



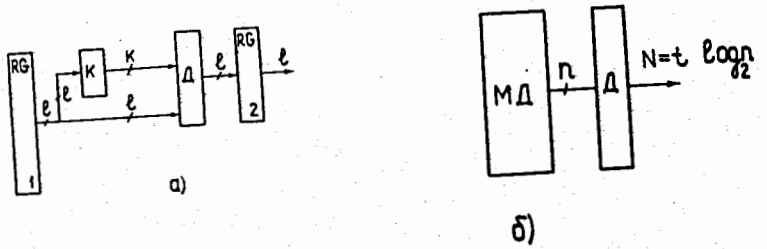



Рис. 8. Многоканальная система передачи
а) с применением избыточного кода, Д-декодер,
б) с использованием синдромного кодирования, МД-многоканальный детектор.

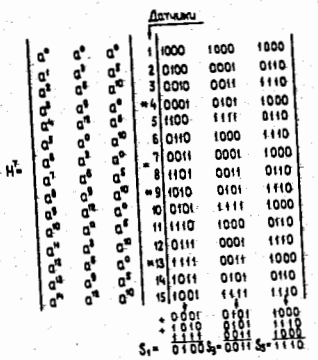



Рис. 9. Матрица проверочных соотношений для БЧХ-кода
$m = 4, n$=I5, $t$= 3.

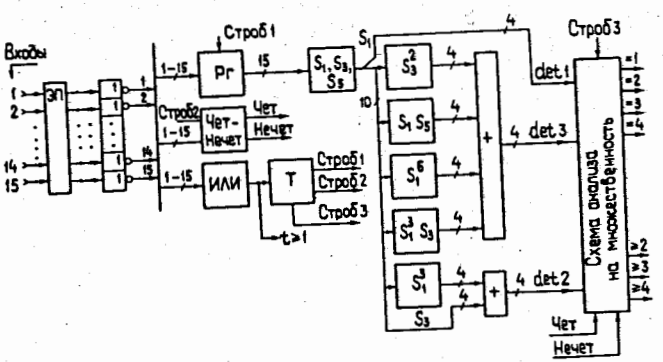



Рис. I0. Блок-схема мажоритарной схемы совпадений.
Т-схема задержки, + - сумматоры по модулю два.

$$X^3 + \sigma_1 X^2 + \sigma_2 X + \sigma_3 = 0, \qquad (5)$$

где $\sigma_1 = S_1 = a$,

$$\sigma_2 = \frac{S_1^2 S_3 + S_5}{S_1^3 + S_3} \quad \text{и} \quad \sigma_3 = \frac{S_1 S_5 + S_3^2 + S_1^3 S_3 + S_1^6}{S_1^3 + S_3}.$$

Далее вычисляем

$$\sigma_2 = a^{14} \quad \text{и} \quad \sigma_3 = a^8.$$

Подставляя значение $\sigma_1$, $\sigma_2$, и $\sigma_3$ в уравнение (5), получим:

$$X^3 + aX^2 + a^{14} X + a^8. \qquad (6)$$

Можно проверить, что элементы $a^3$, $a^8$ и $a^{12}$, соответствующие позициям сработавших сцинтилляторов, являются корнями уравнения (6).

Нетрудно заметить, что такой метод нахождения позиций сработавших сцинтилляторов вряд ли удовлетворяет условию поставленной нами выше задачи. Учитывая тот факт, что синдром представляет собой 10-разрядное слово, и используя возможность выполнения совмещенных операций в поле Галуа, мы попытаемся решить задачу табличным методом с помощью ППЗУ. Прежде всего решим вопрос о количестве сработавших сцинтилляторов.

Алгоритм отбора по количеству базируется на следующем свойстве матрицы $L_t$:

Матрица размерности $t \times t$

$$L_t = \begin{vmatrix} S_1 & 1 & 0 & 0 & \dots & 0 \\ S_3 & S_2 & S_1 & 1 & \dots & 0 \\ S_5 & S_4 & S_3 & S_2 & \dots & 0 \\ \vdots & & & & & \\ S_{2t-1} & S_{2t-2} & S_{2t-3} & S_{2t-4} & \dots & S_t \end{vmatrix}$$

невырождена, если степенные симметрические функции $S_i$ зависят от $t$ или $t+1$ элементов поля, и вырождена, если $S_i$ зависят от меньшего, чем $t-1$, числа различных элементов поля /16/. Практически это значит, что для определения величины $t$ нужно вычислять определитель $L_t$.

Выражения для определителя $L_t$ для $t = 1, 2, \dots, 5$, вычисленные с помощью ЭВМ, приведены в табл. 4.

Каждое из значений определителя вычисляется отдельно и анализи-



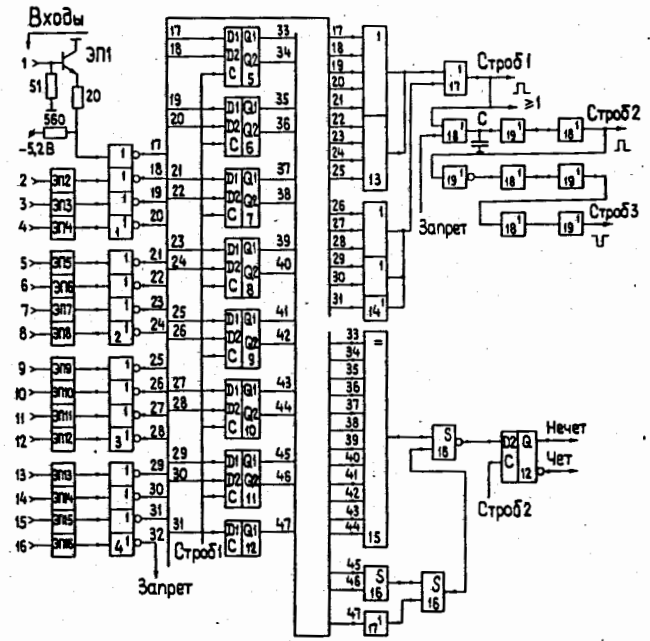



Рис. II. Принципиальная схема входной части.

Таблица 4

| 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 | $S_1^{10} + S_1 S_3 + S_1^5 S_5 + S_1^3 S_7 + S_1^2 S_3 S_5 + S_1 S_3^3 + S_3 S_7 + S_5^2$ |
| 5 | $S_1 S_3^3 S_5 + S_1 S_5 S_9 + S_1^5 S_7 + S_1^2 S_3 S_5^2 + S_1^2 S_3^2 S_7 + S_1^3 S_3 S_9 + S_1^3 S_5 S_7 + S_1^4 S_3^2 S_5 + S_1^5 S_3^7 S_7 + S_1^5 S_5^2 + S_1^6 S_3^3 + S_1^6 S_9 + S_1^7 S_3 S_5 + S_1^8 S_7 + S_1^9 S_3^2 + S_1^{12} S_3 + S_1^{15} + S_3^2 S_9 + S_3^5 S_5^3$ |

руется на "0". В зависимости от результата анализа принимается решение, причем количество слагаемых в каждом из выражений для $L_t$ определяется величиной $t$ и не зависит от числа $n$. Логические уравнения для мажоритарной схемы совпадений, содержащей $n$ входов и $t$ выходов, имеют следующий вид:

Выход I = $det L_1 \vee det L_2 \vee det L_3 \ldots\ldots \vee det L_j \ldots \vee det L_t \geq 1$

Выход 2 = $det L_2 \vee det L_3 \ldots\ldots \vee det L_j \ldots \vee det L_t \geq 2$

Выход 3 = $det L_3 \ldots\ldots \vee det L_j \ldots \vee det L_t \geq 3$ (7)

•

•

•

Выход $j$ = $det L_j \ldots\ldots \vee det L_t \geq j$

•

•

Выход $t$ = $det L_t \geq t$.

Соотношения /7/ описывают блок-схему мажоритарной схемы совпадений, содержащей $n$ входов и $t$ выходов.

На рис. I0 приведена блок-схема предлагаемой мажоритарной схемы совпадений. Предполагается, что входные и выходные сигналы имеют уровни ТТЛМ, а в качестве элементной базы используются быстродействующие микросхемы 500-й серии, у которой логические уровни имеют напряжения - 0,8 В и - I,6 В. Работает схема следующим образом. С помощью эмиттерных повторителей (ЭП) входные сигналы с уровнями ТТЛМ преобразуются в уровни ЭСЛ /- I,6 В/ и после инвертирования с уровнями - 0,8 В поступают на общую магистраль. От магистрали данные поступают на входы трех устройств. На выходе схемы ИЛИ вырабатывается сигнал "быстрое ИЛИ", который идентифицируется как $t \geq I$. После задержки на выходе устройства Т вырабатывается три стробирующих импульса. С помощью импульса "Строб I" происходит занесение входных сигналов на регистр Рг. Импульс "Строб 2" используется для стробирования триггера, на котором запоминается сигнал „Чет" или „Нечет". Как будет показано ниже, эти сигналы служат для расширения диапазона регистрируемой множественности. В принципе наличие регистра в описываемом устройстве не обязательно, и к тому же он вносит дополнительную задержку. Однако использование регистра на входах прибора совместно со стробирующими импульсами позволяет более надежно регистрировать результаты решения. Далее, данные с выходов регистра поступают на устройство вычисления синдрома $S_1$, $S_3$ и $S_5$. Как видно из табл. 4, вычисление определителя первого порядка не требуется, так как он



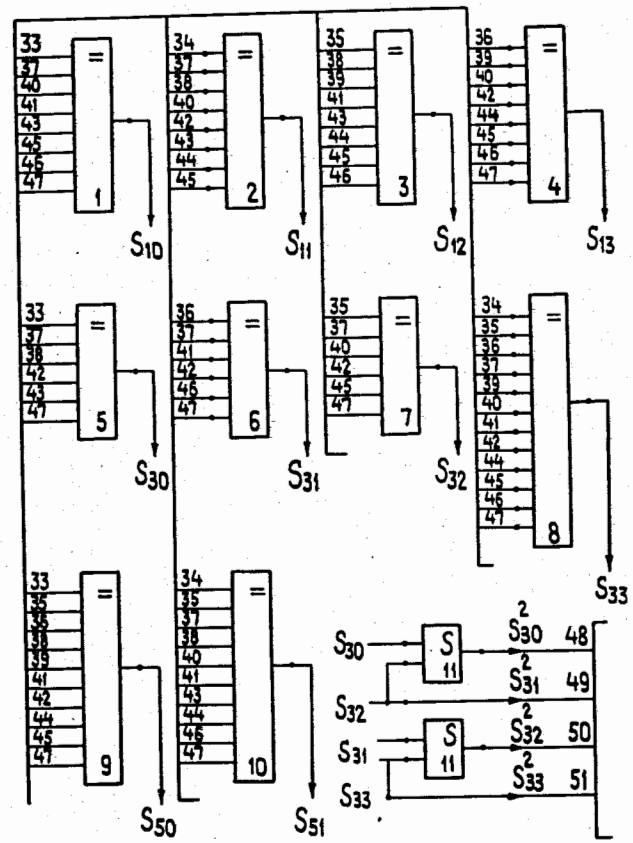



Рис. I2. Принципиальные схемы для формирования синдрома и возведения элемента $S_3$ в квадрат.

равен $S_1$. Для вычисления определителей второго и третьего порядков предварительно с помощью ППЗУ находятся значения $S_3^2$, $S_1 S_5$, $S_1^6$ $S_1^3 S_3$, $S_1^3$, и на сумматорах по модулю два получают окончательные результаты. И, наконец, с помощью схемы анализа на множественность вырабатывают выходные сигналы. Величина задержки сигналов по тракту вход - выход не превышает 35 нс.

Может возникнуть вопрос: " А зачем нужна такая относительно сложная процедура определения множественности?" Суть дела в том , что синдром несет в себе данные не только о количестве сработавших сцинтилляторов, но и содержит информацию об их координатах , в то время как в широко известных устройствах подобного класса имеется возможность регистрировать только множественность сигналов, поступающих на входы.

Теперь рассмотрим принципиальные схемы отдельных узлов процессора. На рис. II приведена схема входной части , которая включает: эмиттерные повторители ЭП1 - ЭП16, инверторы, триггерный регистр, схему "быстрое ИЛИ", построенную на микросхемах I3 - I4,схему регистрации сигналов"Чет"и "Нечет" /микросхемы I2 и I5,- I7 /, а также микросхемы, с помощью которых вырабатываются стробирующие импульсы. Работа устройства может быть запрещена путем подачи сигнала "Запрет". С целью упрощения рисунков здесь и ниже не показаны номера контактов у микросхем, а также нагрузочные резисторы на их выходах. Типичная величина сопротивления нагрузочных резисторов равна 620 Ом. Конденсатор С используется для юстировки величин задержек стробирующих импульсов. С целью упрощения организации связей на всех рисунках принята единая нумерация монтажных входов и выходов.

На рис. I2 приведены принципиальные схемы для формирования синдрома и возведения элемента $S_3^2$ в квадрат в поле $GF(2^4)$. Связи между выходами триггеров регистра и входами схем проверки на четность I - I0 выполнены в соответствии с позициями единиц в матрице $H^T$, изображенной на рис. 9 . Так, для формирования компоненты $S_{10}$ элемента $S_1$ на входы микросхемы I сигналы подаются с выходов тех триггеров регистра, которые имеют номера 33 ,37, 40, 4I, 43 , 45, 46, и 47. Эти цифры, как нетрудно заметить, соответствуют позициям единиц, расположенным в первом столбце матрицы $H^T$.Другими словами, с помощью матрицы $H^T$задаются связи между входными каналами и входами схем проверки на четность. Такие связи в принципе можно запрограммировать и выполнить с помощью ЭВМ. Этот факт лишний раз иллюстрирует, какие широкие потенциальные возможности имеют процессоры с алгебраической структурой. Справа внизу приведена принципиальная схема возведения элемента $S_3$ в квадрат. В принципе для этой цели можно было бы применить ППЗУ, однако ипользование сумматоров по модулю два



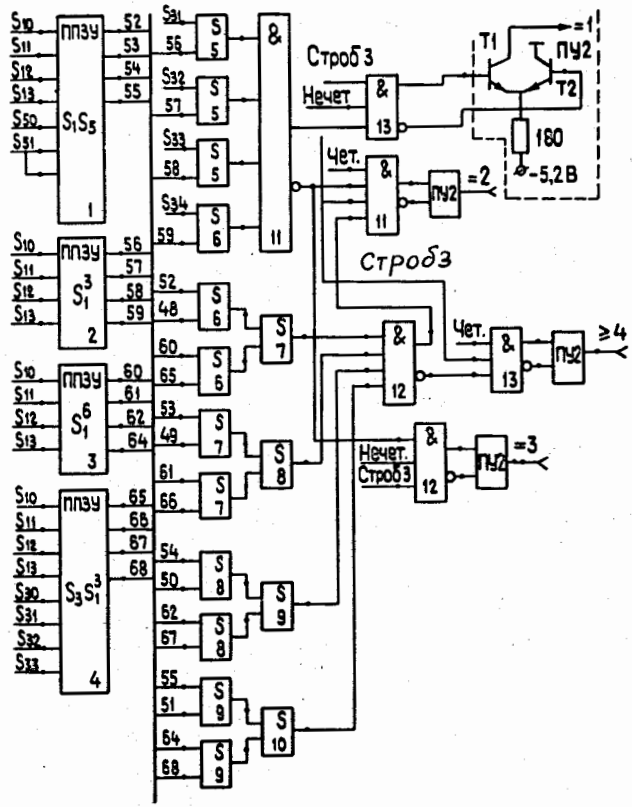



Рис. I3. Схемы для вычисления определителей и анализа входных сигналов на множественность.

в данном конкретном случае является более экономичным решением. На рис. I3 изображены принципиальные схемы вычисления определителей и анализа входных сигналов на множественность. Определители вычисляются с помощью ППЗУ (микросхемы I - 4) и сумматоров по модулю два (микросхемы 5 - 9). Алгоритм отбора сигналов следующий.

Если определитель $det2 = 0$ и есть сигнал „Нечет", то имеет место совпадение импульсов на выходах микросхемы I3 (сверху на рис. I3). На выходах микросхемы I3 импульсы появляются только в том случае, когда на всех ее трех входах сформируются сигналы с уровнем - I,6 В. С помощью преобразователей ПУ2 уровни ЭСЛ трансформируются в уровни NIM на выходах. В конечном итоге принимается решение, что $t = I$. Далее, если $det2 \neq 0$ и есть сигнал „Чет", то считаем, что $t = 2$. При этих же данных, но если сформировался сигнал „Нечет", то $t = 3$. Если определители второго и третьего порядков не равны нулю и есть признак „Чет", то полагаем, что $t \geq 4$.

Для того, чтобы запрограммировать ППЗУ, необходимо составить специальные таблицы. На рис. I4 в качестве примера приведена часть такой таблицы, с помощью которой программируется операция умножения двух элементов A и B в поле $GF(2^4)$. В качестве модуля ППЗУ используется микросхема типа К500РЕI49, которая имеет 8 адресных входов и 4 выхода. Слева на рис. I4 приведены двоичные числа, соответствующие элементам поля, далее представлены их I6-ричные эквиваленты, которые используются в процессе работы с дисплеем. И, наконец, в последней колонке даны коды произведения. Причем здесь принято условно, что старший разряд произведения находится слева.

На рис. I5 изображена схема, с помощью которой можно вырабатывать сигнал на выходе прибора при условии, что сигналы на входах поступили от заданной комбинации сцинтилляторов. Основным компонентом данной схемы является модуль запоминающего устройства с произвольной выборкой емкостью IK (микросхема I0). Адресные входы модуля памяти $A_0$ - $A_9$ соединены с выходами мультиплексоров 5 - 9. Данные на входы мультиплексоров поступают от трех направлений: от тумблерного регистра ТбI - Тб10, от I0-разрядного счетчика (микросхемы 2 - 4) и от схемы формирования синдрома. С помощью тумблерного регистра имеется возможность вручную в заданные ячейки памяти записать единицы. В результате на выходе модуля памяти сигнал будет появляться только в тех случаях, когда код синдрома будет равен адресам тех ячеек памяти, в которые были предварительно записаны единицы. С помощью счетчика и генератора импульсов (микросхема I) в ячейки памяти предварительно записываются нули. Тумблер I3 используется для задания необходимого режима работы счетчика. Тумблеры



| N | A | B | A | B | Код |
|---|---|---|---|---|---|
| 0 | 0000 | 0000 | 0 | 0 | 0 0 |
| 1 | 0000 | 0001 | 0 | 1 | 0 0 |
| 2 | 0000 | 0010 | 0 | 2 | 0 0 |
| 3 | 0000 | 0011 | 0 | 3 | 0 0 |
| : | : : : : | | : | : | : : |
| 15 | 0000 | 1111 | 0 | F | 0 0 |
| 16 | 0001 | 0000 | 1 | 0 | 0 0 |
| 17 | 0001 | 0001 | 1 | 1 | 21 |
| 18 | 0001 | 0010 | 1 | 2 | 42 |
| 19 | 0001 | 0011 | 1 | 3 | 41 |
| 20 | 0001 | 0100 | 1 | 4 | 84 |
| 21 | 0001 | 0101 | 1 | 5 | 84 21 |
| 22 | 0001 | 0110 | 1 | 6 | 8 2 |
| 23 | 0001 | 0110 | 1 | 7 | 8 1 |
| 24 | 0001 | 1000 | 1 | 8 | 1 |
| 25 | 0001 | 1001 | 1 | 9 | 2 |
| 26 | 0001 | 1010 | 1 | A | 42 1 |
| 27 | 0001 | 1011 | 1 | B | 4 |

Рис. I4. Пример таблицы, используемой для программирования ППЗУ.

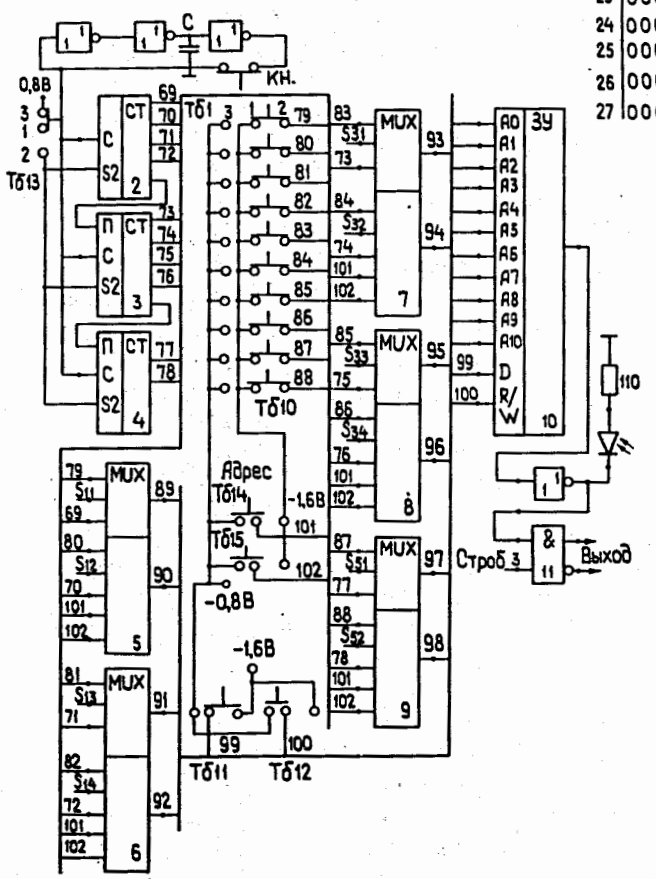



Рис. I5. Схема выработки сигнала на выходе при условии появления заданной комбинации импульсов на входах.

Тб14 - Тб15 необходимы для установки кодов адресов входных каналов мультиплексоров. С помощью тумблера Тб11 задается код, необходимый для записи. Тумблер Тб12 используется для установки режима "Запись-чтение".

Для дешифровки координат сработавших сцинтилляторов можно использовать модуль ППЗУ.

Таким образом, поставленная нами выше задача решена с помощью специализированного процессора, имеющего алгебраическую структуру. Поскольку информация предварительно сжимается, то в таком случае появляется возможность использовать для решения быстродействующие модули памяти. В тех случаях, когда требуется построить процессор, имеющий большее число входов, необходимо выбрать неприводимый полином более высокой степени. Например, при $n = 31$ таким полиномом будет $X^5 + X^2 + 1$. Конечно, в этом случае необходимы и модули памяти большей емкости. Однако технология БИС бурно развивается, и поэтому в ближайшее время следует ожидать появления модулей ППЗУ емкостью 64К и более.

## 7. Расчет переключательных функций

Расчет и минимизация переключательных функций (ПФ) необходимы для того, чтобы по полученным булевым выражениям создавать различные логические схемы. Однако при большом числе переменных, когда $m > 3$, широко известные методы расчета и минимизации (ПФ) не носят формального характера и скорее похожи на искусство, нежели на точный алгоритм. В работе /5/ рассматриваются вопросы расчета ПФ с помощью ЭВМ. Причем отличительной особенностью метода является то, что входные и выходные переменные являются элементами поля Галуа $GF(2^m)$, что позволяет любую ПФ $m$ переменных представить в виде полинома $2^m - 1$ степени. Перспективность такого направления синтеза ПФ заключается в том, что для их расчета можно использовать современные мощные ЭВМ и аналитические методы.

Суть дела в том, что теория поля Галуа является естественным продолжением булева поля. Однако представление ПФ в виде элементов поля Галуа $GF(2^m)$ (галуа-переключательные функции - ГПФ) имеет ряд преимуществ:

- над ГПФ можно выполнять алгебраические операции, что упрощает проблему минимизации и ее формального представления;
- поскольку состояние входов и выходов комбинационной схемы или последовательностного автомата кодируются элементами поля Галуа, то следующее состояние можно представить как полиномиальную функцию текущего состояния и текущего выхода;



- представление ГПФ в виде полинома, в котором как коэффициенты, так и переменные являются элементами поля Галуа, при большом числе переменных позволяет использовать стандартные и вновь создаваемые системы программирования для расчета довольно сложных устройств дискретной логики;
- представление ПФ в виде полинома имеет еще и то преимущество, что описание многозначных и многоуровневых схем имеет весьма компактный вид.

Фундаментальные свойства ГПФ рассмотрены в работе (5). Известно, что любую ПФ $F(X) = F(X_0, X_1, \ldots X_{m-1})$ $m$ аргументов можно представить в виде полинома

$$F(X_0, X_1, \ldots X_{m-1}) = B(0) + A(1)X + A(2)X^2 + \ldots + A(2^m - 1)X^{2^m - 1}. \quad (8)$$

Здесь и ниже знак + будет обозначать сумму по модулю два, а коэффициенты $A(K)$ вычисляются из выражения:

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

где $B(a_j)$ - элементы подстановки, которые берутся из таблиц соответствия входов и выходов, $B(0)$ - значение функции на нуле.

Таким образом, для получения алгебраического выражения, с помощью которого реализуется необходимая схема, заданная таблицей, достаточно выполнить следующие процедуры:

- по выбранному неприводимому полиному находят все ненулевые элементы поля Галуа $GF(2^m)$;
- вычисляют коэффициенты $A(K)$, предварительно задавшись таблицами соответствия входов и выходов;
- проводится разложение коэффициентов $A(K)$ и степеней $X$ по базисным элементам;
- приводятся подобные члены.

Рассмотрим примеры.

Пример 1. Рассчитаем схему одноразрядного сумматора на три входа. Вначале составим, как это и принято, таблицу соответствия входов и выходов (табл. 5). Слева в таблице показана последовательность элементов поля $GF(2^4)$ и их двоичные эквиваленты. Справа располагаются те значения, которые необходимо получить на выходах создаваемого устройства. Нетрудно заметить, что с помощью табл. 5 описывается работа обычного комбинационного сумматора на три входа и два выхода (сумма и перенос в следующий разряд). Работа такого сумматора задается табл. 6. Здесь $X_0$, $X_1$, $X_2$ - значения первого, второго слагаемых и входа переноса, С и П - значения суммы и пере-

Таблица 5

| $X = X_0X_1X_2$/ входы | $B(a_j)$ / выходы |
|---|---|
| 0 = 000 | 0 = 000 00 |
| $a^0$ = I00 | $a^0$ = I00 I0 |
| a = 0I0 | $a^0$ = I00 I0 |
| $a^2$ = 00I | $a^0$ = I00 I0 |
| $a^3$ = II0 | a = 0I0 0I |
| $a^4$ = 0II | a = 0I0 0I |
| $a^5$ = III | $a^3$ = II0 II |
| $a^6$ = I0I | a = 0I0 0I |

Таблица 6

| $X_0X_1X_2$/ входы | С П / выходы |
|---|---|
| 000 | 0 0 |
| I00 | I 0 |
| 0I0 | I 0 |
| 00I | I 0 |
| II0 | 0 I |
| 0II | 0 I |
| III | I I |
| I0I | 0 I |

Таблица 7

| Входы $X = \{X_0X_1X_2X_3\}$ | Выходы $F(X)$ |
|---|---|
| 0 = 0000 | 0 = 0000 |
| $a^0$ = I000 | a = 0I00 |
| a = 0I00 | 0 = 0000 |
| $a^2$ = 00I0 | $a^7$ = II0I |
| $a^3$ = 000I | $a^5$ = 0II0 |
| $a^4$ = II00 | $a^5$ = 0II0 |
| $a^5$ = 0II0 | $a^{11}$ = 0III |
| $a^6$ = 00II | $a^{13}$ = I0II |
| $a^7$ = II0I | $a^0$ = I000 |
| $a^8$ = I0I0 | $a^3$ = 000I |
| $a^9$ = 0I0I | $a^{14}$ = I00I |
| $a^{10}$ = III0 | $a^{14}$ = I00I |
| $a^{11}$ = 0III | 0 = 0000 |
| $a^{12}$ = IIII | $a^2$ = 00I0 |
| $a^{13}$ = II0II | $a^4$ = II00 |
| $a^{14}$ = I00I | $a^0$ = I000 |



носа на выходах сумматора. Для нашего примера, как это видно из таблицы 5, элементы подстановки B(I), B(2), B(3), B(4), B(5), B(6), и B(7) представляют собой элементы поля $a^0$, $a^0$, $a^0$, $a^1$, $a^3$, $a^1$ соответственно. Тогда численное значение коэффициента A(I) получается из выражения

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

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

$$= a^0 + a^6 + a^5 + a^5 + a^4 + a^5 + a^2 = a^0 + a^6 + a^4 + a^5 + a^2$$

$= a^0 = I00$. Здесь операция деления двух элементов заменена операцией умножения элемента A на инверсный элемент B .
Далее имеем:

$$A(2) = \frac{a^0}{(a^0)^2}+\frac{a^0}{(a^1)^2}+\frac{a^0}{(a^2)^2}+\frac{a^1}{(a^3)^2}+\frac{a^1}{(a^4)^2}+\frac{a^3}{(a^5)^2}+\frac{a^1}{(a^6)^2} = a = 0I0 .$$

Аналогичные вычисления дают:

A(3) = a = I00, A(4) = A(7) = 0, A(5) = a = 0II, A(6) = a = I0I.
С учетом проведенных вычислений выражение ( 8 ) имеет вид:

$$F(X) = X + a X^2 + X^3 + a^4 X^5 + a^6 X^6 \qquad (9) .$$

В принципе с помощью выражения ( 9 ) можно построить схему одноразрядного сумматора. Однако такая схема была бы слишком громоздкой, т. к. для ее реализации потребовались бы схемы возведения в степень и схемы умножения двух элементов в поле Галуа $GF(2^3)$. Поэтому выражение ( 9 ) целесообразно упростить, для чего достаточно коэффициенты $a^4$, $a^6$ и переменные X , $X^2$, $X^3$, $X^5$, $X^6$ представить в виде полиномов и разложить по базисным элементам. Исходя из этого, имеем:

$$F(X) = (X_0a^0 + X_1a + X_2a) + a\left[X_0 + X_2a + (X_1 + X_2)a^2\right] +$$

$$(X_0 + X_1 + X_2 + X_1X_2) + (X_1 + X_0X_1 + X_0X_2)\,a + (X_2 + X_0X_1)\,a^2]$$

$$+ (a + a^2)\,(X_0 + X_1 + X_2 + X_1X_2) + (X_1 + X_2 + X_0X_2)\,a + (X_1 + X_0X_1 + X_0X_2)a^2$$

$$+ (a^0 + a^2)\,[(X_0 + X_1 + X_2 + X_1X_2) + (X_2 + X_0X_1)\,a + (X_1 + X_2 + X_0X_2)\,a^2].$$

После умножения и приведения подобных членов получим следующие булевы выражения, с помощью которых описывается работа полного одноразрядного сумматора :

Сумма $C = X_0 + X_1 + X_2$ и перенос $\Pi = X_0X_1 + X_0X_2 + X_1X_2$.

Оба равенства реализуются при помощи двух полусумматоров ПС1 и ПС2 (рис. I6) . Такая схема получается, если для построения полусумматора использовать известную схему, состоящую из логических элементов И, ИЛИ и НЕ. Однако следует отметить, что в связи с развитием техники интегральных микросхем при создании устройств дискретной логики все более широкое применение находит логический базис "Исключающее ИЛИ" и элемент „И".

Может возникнуть вопрос: "Какой смысл рассчитывать уже известные схемы ?" Дело в том, что , когда развивается новая методика, очень важно проверить ее работоспособность с помощью уже известных тестов. Кроме того, необходимо создать соответствующее математическое обеспечение . И, наконец, возможность использования ЭВМ для расчета сколь угодно сложных устройств дискретной логики открывает широкие перспективы для комплексной автоматизации проектирования микропроцессоров и микро-ЭВМ.

Пример 2. Рассчитаем схему последовательностного автомата, на вход которого в заданные моменты времени последовательно подаются в порядке возрастания степеней элементы поля Галуа $GF(2^4)$, образованные над неприводимым полиномом $X^4 + X + I$, а на выходе получаются элементы этого же поля , но в заданной последовательности, как это видно из таблицы соответствия входов и выходов (табл. 6). Последовательность элементов поля $GF(2^4)$ в порядке возрастания их степеней получается довольно просто с помощью счетчика в поле $GF(2^4)$, который представляет собой сдвиговый регистр с логическими обратными связями. Если в младший разряд такого регистра поместить единицу, а в остальные разряды - нули, то последовательные сдвиги дают представление последовательных степеней элемента А, корня многочлена $X^4 + X + I$, в такой же точности, в какой они приведены в табл. 7 слева, причем произвольный элемент А в поле $GF(2^4)$ имеет вид: $A_0a^0 + A_1a^1 + A_2a^2 + A_3a^3$. Для построения схемы, которая выполняла бы по существу функцию преобразования 4-разрядных кодов, необходимо прежде всего вычислить значения I6 коэффициентов в полиномиальном представлении 4 переменных:

$$F(X_0, X_1, X_2, X_3) = A(0) + A(1)X + A(2)X^2 + A(3)X^3 + A(4)X^4 + A(5)X^5 + A(6)X^6 + A(7)X^7 + A(8)X^8 + A(9)X^9 + A(10)X^{10} + A(11)X^{11} + A(12)X^{12} + A(13)X^{13} + A(14)X^{14} + A(15)X^{15}$$

После вычисления на ЭВМ получились следующие выражения, разложенные по базисным элементам (см. приложение) :



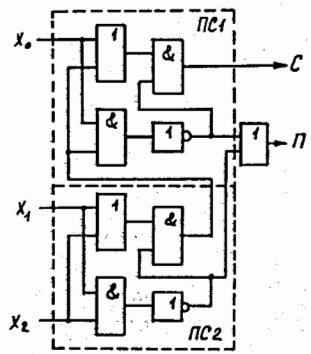



Рис.I6. Схема одноразрядного комбинационного сумматора.
ПС1 и ПС2 - полусумматоры.

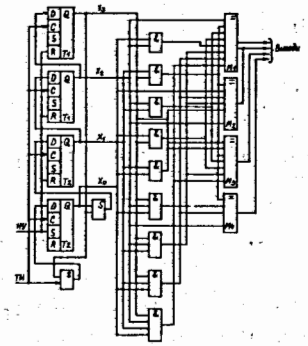


Рис. I7. Схема последовательностного автомата. МI-М4 - микросхемы КI55ИП2. НУ - начальная установка.

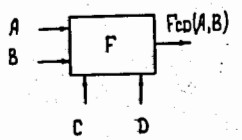
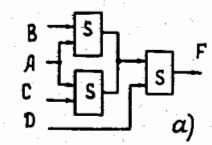



Рис. I8. Блок-схема УДПЛМ. А,В - входы для переменных, С,D - входы для коэффициентов настройки.




| C | D | F |
|---|---|---|
| 0 | 0 | $\overline{A \vee B}$ |
| 0 | I | $\overline{A \vee B}$ |
| I | 0 | $\overline{AB}$ |
| I | I | AB |
| A(B) | 0 | A + B |
| A(B) | I | A = B |

б)

Рис. I9. Схема модуля и его таблица истинности. + - сумма по модулю два, = - логическая равнозначность.

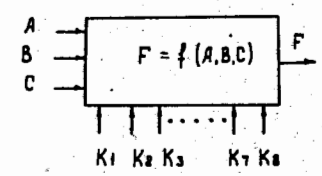



Рис. 20. Блок-схема программируемого модуля на основе мультиплексора.

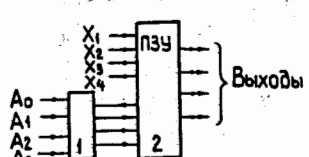



Рис. 22. Схема, реализующая возведение в степень с одновременным умножением.

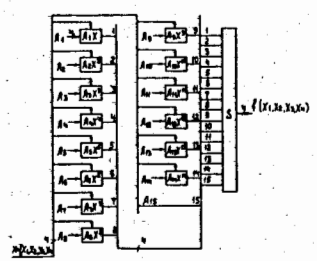


Рис. 2I. Блок- схема предлагаемого УДПЛМ.

$X_1 + X_0X_2 + X_0X_3 + X_1X_2 + X_1X_3 + X_0X_1X_3 + X_1X_2X_3 + X_0X_1X_2X_3$ <a⁰>

$X_0 + X_2 + X_3 + X_1X_3 + X_0X_1X_3 + X_1X_2X_3$ <a>

$X_3 + X_0X_1 + X_0X_3 + X_1X_2 + X_1X_3 + X_1X_2X_3 + X_0X_1X_2X_3$ <a²>

$X_2 + X_1X_3 + X_0X_1X_3 + X_0X_2X_3$ <a³>

Все необходимые вычисления для примера 2 выполнены на ЭВМ ЕС-1033. На рис. 17 приведена принципиальная схема, в которой используются обычные микросхемы, выпускаемые промышленностью. Можно проверить, что данная схема генерирует последовательность слов в соответствии с табл. 7. Такие схемы могут найти применение, например, в устройствах с микропрограммным управлением для получения заданной последовательности двоичных слов.

## 8. Быстрые программируемые логические модули

Техника современного эксперимента требует совершенствования логических схем отбора как по быстродействию, так и по числу каналов регистрации. Весьма актуальной является также задача создания таких логических устройств, в которых имелась бы возможность быстрого перепрограммирования на выполнение различного рода задач, возникающих в процессе эксперимента. В последнее время как в теоретическом, так и в практическом аспекте очень интенсивно ведутся исследования по разработке универсальных динамически программируемых логических модулей (УДПЛМ). На рис. 18 приведена блок-схема УДПЛМ. Она содержит два входа для переменных A и B и два входа C и D, на которые подаются коэффициенты настройки. Меняя эти коэффициенты, можно быстро изменять тип выполняемой булевой функции F. На рис. 19а изображена принципиальная схема такого модуля. Функции, выполняемые таким модулем, приведены на рис 19б. Основная проблема заключается в том, чтобы, используя минимальное число входов настройки, создать модуль, выполняющий максимальное число логических функций при минимальном числе используемых логических элементов. В ядерной электронике вопросам создания УДПЛМ уделяется серьезное внимание. В /17/ описан управляемый логический модуль, с помощью которого можно получить минтермы 4 переменных. Он состоит из 16 элементов И, каждый из которых имеет 6 входов: 4 используются для подачи сигналов, соответствующих четырем переменным, а на остальные подаются сигнал управления и стробирующий импульс. Выходы всех элементов И объединяются с помощью элемента ИЛИ. Коэффициенты настройки (управление) могут подаваться как от тумблерного регистра вручную, так и от триггерного регистра. Как нетрудно заметить, функциональные возможности такого модуля весьма ограничены. Применение микросхем средней и большой степени интеграции позволяет существенно расширить диапазон выполняемых функций и повысить эффективность использования. В /18/ приведена схема программируемого модуля, созданного на основе 8-входового мультиплексора (рис. 20). В таком модуле переменные A, B и C подаются на три адресных входа, а восемь информационных входов служат для подачи коэффициентов настройки $K_1 - K_8$. Один модуль обеспечивает реализацию 256 различных ПФ при задержке 5 нс. Используя две такие схемы, можно построить модуль на 5 входов.

Широко известны также работы, в которых для реализации ПФ используются ПЛМ, ЗУПВ или ППЗУ. В таких случаях входные переменные подаются на адресные входы памяти, на выходе которой можно получить $2^N$ комбинаций от $N$ входных переменных.

Выше нами были рассмотрены преимущества, которые дает применение ГПФ для синтеза и построения логических схем. Здесь мы отметим еще одно важное обстоятельство, что элементы поля можно рассматривать как минтермы $m$ переменных. Например, элемент поля $GF(2^4)$ можно представить как $a^8 = 1010 = X\bar{X}X\bar{X}$. Напомним, что ГПФ 4 переменных $X_1X_2X_3X_4$ можно представить в виде полинома 15-й степени:

$$F(X) = F(X_1X_2X_3X_4) = A(0) + A(1)X + A(2)X^2 + \ldots + A(14)X^{14} + A(15)X^{15} \quad (10)$$

где $X = \{X_1X_2X_3X_4\}$ - входные переменные, A(0) - значение функции в нулевой точке, A(1), A(2), .. A(14) и A(15) - коэффициенты настройки. Тип схемы, реализуемой с помощью ГПФ, определяется значениями коэффициентов A(1) ÷ A(15). Другими словами, меняя быстро коэффициенты настройки в устройстве, с помощью которого реализуется равенство (10), можно динамически перестраивать работу такого устройства на выполнение наперед заданных функций.

Теперь остается определить, какие аппаратные средства для этого требуются. Как видно из выражения (10), для построения устройства, с помощью которого можно было бы вычислять полиномиальное представление ПФ, необходимы схемы для выполнения операций умножения, возведения в степень и сложения элементов поля Галуа. Каким образом такие операции можно эффективно выполнять, мы рассмотрели в предыдущих разделах. На рис. 21 приведена блок-схема УДПЛМ, которая построена в соответствии с равенством (10). Модуль имеет 4 входа для переменных, 60 входов для коэффициентов настройки и 4 выхода. В соответствии с теорией, такой модуль может выполнять 65536 различных переключательных функций ! Причем коэффициенты настройки можно

подавать в случае необходимости довольно быстро с помощью ЭВМ. Для практической реализации такого устройства необходимо иметь 15 схем для одновременного умножения и возведения в степень двух элементов в поле $GF(2^4)$. Если для этих целей использовать быстродействующие ППЗУ типа К500РЕ149, то такие схемы получаются однотипными и довольно простыми. Разница только заключается в содержимом ППЗУ. На рис.22 приведена такая схема. Она состоит из 4-разрядного регистра I, на котором хранятся коэффициенты настройки модуля ППЗУ. Рассмотрим примеры, которые иллюстрируются с помощью табл. 8. В первой колонке слева приведены элементы поля $GF(2^4)$ и их двоичные эквиваленты. Сигналы, соответствующие этим кодам, подаются на 4 входа модуля. В следующих колонках даны соответствующие им значения, которые должны получаться на выходах модуля, и вычисленные на ЭВМ коэффициенты А(К). Полагаем, что функция на выходах модуля имеет истинное значение, если она равна единичному элементу $a^0 = 1000$. Вычисленные на ЭВМ коэффициенты для настройки модуля на выполнение ПФ совпадения равны ($A(1) = a^3$, $A(2) = a^6$, $A(3) = a^9$, $A(4) = a^{12}$, $A(5) = a^0$, $A(6) = a^3$, $A(7) = a^6$, $A(8) = a^9$, $A(9) = a^{12}$, $A(10) = a^0$, $A(11) = a^3$, $A(12) = a^6$, $A(13) = a^9$, $A(14) = a^{12}$, $A(15) = a^0$. Подставляя их в равенство (10), получим:

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

Тот факт, что с помощью выражения (11) описывается работа схемы совпадения на 4 входа, можно проверить двумя способами:

1) Подставим значение $X = a^{12}$ и после вычисления (например, $a^3 a^{12} = a^{15} = a^0$, $a^6 (a^{12})^2 = a^{30} = a^{15} a^{15} = a^0$ и т. д.) получим 15 слагаемых, каждый из которых равен $a^0$. При суммировании таких членов по модулю два получим значение $a^0$.

2) Если выражение (11) упростить, разложив элементы поля различных степеней по базисным элементам, то после упрощения с помощью ЭВМ на цифропечать выдается следующее равенство:

$$F(X) = (X_1X_2X_3X_4)a^0.$$

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

Рассмотрим вторую колонку, с помощью которой иллюстрируется настройка модуля на выполнение "пассивной" операции, когда коды на входах и выходах одинаковы (повторение логических сигналов). Другими словами, мы тем самым электрически выключаем модуль из системы. Как показывают расчеты на ЭВМ, в этом случае только один коэффициент при X не равен нулю.



Таблица 8

Представление логических функций элементами поля Галуа $GF(2^4)$

| Функция / Входы | Совпадение | | Повтор | | Инверсия | |
|---|---|---|---|---|---|---|
| $X = X_1X_2X_3X_4$ | $F = X_1X_2X_3X_4$ | A(K) | $F = X$ | A(K) | $F = X_1X_2X_3X_4$ | A(K) |
| $a^1$=0100 | 0000 | $a^3$ | 0100 | $a^0$ | 1011 | $a^0$ |
| $a^2$=0010 | 0000 | $a^6$ | 0010 | 0 | 1101 | 0 |
| $a^3$=0001 | 0000 | $a^9$ | 0001 | 0 | 1110 | 0 |
| $a^4$=1100 | 0000 | $a^{12}$ | 1100 | 0 | 0011 | 0 |
| $a^5$=0110 | 0000 | $a^0$ | 0110 | 0 | 1001 | 0 |
| $a^6$=0011 | 0000 | $a^3$ | 0011 | 0 | 1100 | 0 |
| $a^7$=1101 | 0000 | $a^6$ | 1101 | 0 | 0010 | 0 |
| $a^8$=1010 | 0000 | $a^9$ | 1010 | 0 | 0101 | 0 |
| $a^9$=0101 | 0000 | $a^{12}$ | 0101 | 0 | 1010 | 0 |
| $a^{10}$=1110 | 0000 | $a^0$ | 1110 | 0 | 0001 | 0 |
| $a^{11}$=0111 | 0000 | $a^3$ | 0111 | 0 | 1000 | 0 |
| $a^{12}$=1111 | 1000 | $a^6$ | 1111 | 0 | 0000 | 0 |
| $a^{13}$=1011 | 0000 | $a^9$ | 1011 | 0 | 0100 | 0 |
| $a^{14}$=1001 | 0000 | $a^{12}$ | 1001 | 0 | 0110 | 0 |
| $a^{15}$=1000= | 0000 | $a^0$ | 1000 | 0 | 0111 | a |
| $=a^0=$ | 0000 | | 1000 | 0 | 0111 | $a^{12}$ |
| 0=0000 | 0000 | | 0000 | 0 | 0000 | 0 |

Имеем:

$F(X) = a^0X = a^0(a^0X_1 + a^1X_2 + a^2X_3 + a^3X_4) = a^0X_1 + a^1X_2 + a^2X_3 + a^3X_4 = X$, так как умножение на единичный элемент оставляет множитель без изменения.

Если же в уравнение (11) подставить коэффициенты $a^0$ при X и $a^{12}$ при $X^{15}$, то получим логическое уравнение для инверсии за исключением нулевой точки

$$F(X) = a^0X + a^{12}X^{15} = X + a^{12}.$$ Проверка, например, дает, что при $X = a^{12}$, $F(X) = 0 = 0000$.

Следует отметить, что сам по себе такой универсальный модуль вряд

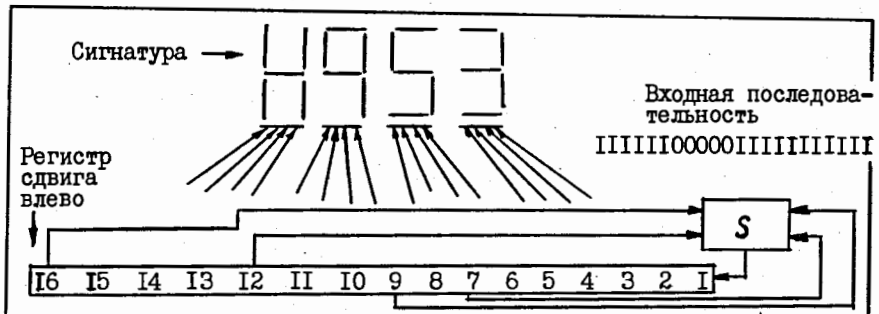
ли с экономической точки зрения может конкурировать с известным набором специализированных логических модулей. Однако стоимость интегральных микросхем непрерывно уменьшается. Более эффективно предложенный способ построения УДЛМ можно было бы использовать, если изготовить набор специализированных микросхем, с помощью которых можно было бы реализовать совмещенные операции в поле Галуа. В то же время использование набора таких однотипных быстродействующих модулей открывает возможности для быстрого перепрограммирования с помощью микропроцессора работы триггерных систем наносекундного диапозона без применения внешних связей, которые в настоящее время являются основными источниками ненадежности и дополнительных задержек в цепях передачи сигналов. Очень важным является установление того факта, что, используя алгебраическую теорию ГПФ, мы имеем возможность, применяя аналитические преобразования и вычисления на ЭВМ, создавать различные устройства дискретной логики с наперед заданными свойствами.

## 9. Синдромное тестирование

Перспективность и огромную практическую ценность алгебраических методов обработки сигналов можно проиллюстрировать на примере бурного развития такого важного направления, как синдромное тестирование, или сигнатурный анализ. Ипользуемые ранее разработанные методы обнаружения и локализации неисправностей в цифровой аппаратуре не удовлетворяют требованиям сегодняшнего дня в связи с массовым выпуском БИС, микропроцессоров и различной измерительной техники со встроенными БИС и микропроцессорами.

Метод синдромного тестирования базируется на свойстве избыточных кодовых последовательностей с помощью кодирующих и декодирующих устройств обнаруживать и исправлять ошибочные символы , которые могут возникать в процессе передачи и приема такой последовательности. Очевидно, что мы имеем здесь прямую аналогию с методом синдромного кодирования, рассмотренного выше. Разница заключается в том, что передаваемое кодовое слово рассматривается не как нулевое слово, а как обычное избыточное информационное слово, которое заранее известно. Если в процессе передачи стандартной последовательности ошибки не произошло, то в декодере должен сформироваться нулевой синдром. В противном случае синдром будет отличен от нуля, и таким образом будет обнаружено искажение эталонной последовательности. Для эффективного использования синдромного тестирования необходимо предусмотреть в контролируемой аппаратуре дополнительные схемы для генерации в заданных точках эталонных последовательностей, а также контакты или специальные разъемы для подключения сигнатурного ана-






| | Содержимое сдвигового регистра | | | | | | | | | | | | | | | | Вход регистра | Вход данных |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | I | I |
| I | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | I | I | I |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | I | I | I | I |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | I | I | I | I | I |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | I | I | I | I | I | I |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | I | I | I | I | I | I | I |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | I | I | I | I | I | I | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | I | I | I | I | I | I | 0 | I | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | I | I | I | I | I | I | 0 | I | I | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | I | I | I | I | I | I | 0 | I | I | 0 | 0 |
| I0 | 0 | 0 | 0 | 0 | 0 | 0 | I | I | I | I | I | I | 0 | I | I | 0 | 0 | 0 |
| II | 0 | 0 | 0 | 0 | 0 | I | I | I | I | I | I | 0 | I | I | 0 | 0 | I | I |
| I2 | 0 | 0 | 0 | 0 | I | I | I | I | I | I | 0 | I | I | 0 | 0 | I | 0 | I |
| I3 | 0 | 0 | 0 | I | I | I | I | I | I | 0 | I | I | 0 | 0 | I | 0 | I | I |
| I4 | 0 | 0 | I | I | I | I | I | I | 0 | I | I | 0 | 0 | I | 0 | I | 0 | I |
| I5 | 0 | I | I | I | I | I | I | 0 | I | I | 0 | 0 | I | 0 | I | 0 | I | I |
| I6 | I | I | I | I | I | I | 0 | I | I | 0 | 0 | I | 0 | I | 0 | I | 0 | I |
| I7 | I | I | I | I | I | 0 | I | I | 0 | 0 | I | 0 | I | 0 | I | 0 | 0 | I |
| I8 | I | I | I | I | 0 | I | I | 0 | 0 | I | 0 | I | 0 | I | 0 | 0 | I | I |
| I9 | I | I | I | 0 | I | I | 0 | 0 | I | 0 | I | 0 | I | 0 | 0 | I | I | I |
| 0 | I | I | 0 | I | I | 0 | 0 | I | 0 | I | 0 | I | 0 | 0 | I | I | – | – |

Рис. 23. К пояснению метода сигнатурного анализа
*S* -сумматор по модулю два.

лизатора. Использование сигнатурного анализа предполагает также разработку новой формы документации для цифровых схем. Так, на принципиальных схемах аналоговых устройств приводятся уровни напряжений и осциллограммы сигналов в различных точках, что помогает обслуживающему персоналу при ремонте и профилактике. Однако в цифровых схемах все входные и выходные сигналы имеют фиксированные уровни и в этом смысле похожи друг на друга. Поэтому на принципиальных схемах в таких случаях приводятся шестнадцатиричные сигнатуры (ключевые коды), с помощью которых однозначно определяются двоичные последовательности , характерные для каждой контрольной точки.

Основным компонентом сигнатурного анализатора является сдвиговый регистр с линейной логической обратной связью (рис. 23). Мы уже рассматривали примеры таких регистров, которые выполняют функции счетчиков в поле Галуа. Однако в случае сигнатурного анализа на вход регистра поступает результат суммирования по модулю два символов входной последовательности и символов, взятых после семи, девяти , двенадцати и шестнадцати элементов задержки. Поскольку регистр сдвига с сумматором по модулю два в цепи обратной связи является линейной цепью, то каждый символ входной последовательности имеет тот же вес, что и символы, поступающие на вход сумматора по модулю два. Причем регистр синхронизируется теми же тактовыми сигналами, что и обрабатываемая последовательность. Важным является тот факт, что входные последовательности могут быть любой длины, но в конце обработки анализируется только код, оставшийся в регистре. Другими словами, любая ошибка в одном или нескольких битах (что зависит от типа используемого кода) независимо от того, когда она произошла, всегда будет определена после остановки регистра в любой момент времени путем сравнения оставшихся символов (шестнадцати в нашем примере) с той комбинацией, которая должна быть. Эти I6 бит, представленные в шестнадцатиричной форме, и являются "сигнатурой обработанной (измеренной) двоичной последовательности". Такой формат выбран с целью облегчения восприятия данных, отображаемых с помощью семисегментных индикаторов.

В таблице на рис. 23 показано, как регистр формирует сигнатуру для 20-бит последовательности IIIIII00000IIIIIIII. Вначале (такты от 0 до 7) схема работает как обычный сдвиговый регистр. На такте 7 первая единица входной последовательности доходит до первого отвода обратной связи . Эта единица по цепи обратной связи попадает на вход сумматора по модулю два и складывается с очередным битом последовательности, имеющим нулевое значение, вследствие этого при следующем синхроимпульсе (в такте 8) в регистр попадает I, а не 0. Так



продолжается до конца обработки, когда 20-бит входная последовательность преобразуется в I6-бит остаток IIOIIOOIOIOIOOII (такт 20). Причем остаток , как правило, отличается от исходной последовательности. В качестве генераторного полинома для построения сигнатурного анализатора фирма Хьюлетт-Паккард выбрала следующий полином :

$$P(X) = X^{16} + X^{12} + X^{9} + X^{7} + I.$$

Сигнатурный анализатор фирмы (7) типа 5004A обеспечивает I00%- ное обнаружение ошибок в один символ входной последовательности и 99,998% при нескольких ошибках. Максимальная частота синхронизации составляет I0 МГц, а минимальная длительность импульса синхронизации - 50 нс.

## I0. Элементы теории корректирующих кодов

### Общие вопросы.

Корректирующие коды или коды, исправляющие ошибки, являются эффективным средством борьбы с помехами в процессе передачи и обработки информации. Коды появились еще в глубокой древности в виде криптограмм, когда ими пользовались для засекречивания важного сообщения от тех, кому оно не было предназначено. Однако математическая теория корректирующих кодов появилась сравнительно недавно- в начале 40-х годов текущего столетия.

Как известно, с помощью $n$ двоичных символов можно получить $2^n$ различных кодовых комбинаций, которые могут быть использованы для кодирования информации. В этом случае они образуют безызбыточный код, называемый натуральным или обычным двоичным кодом. В таком коде любое искажение символов переводит одну кодовую комбинацию в другую, принадлежащую этому классу, и поэтому искажение не может быть обнаружено анализом полученной комбинации. Поэтому корректирующими кодами называются коды, позволяющие обнаруживать и исправлять ошибки, происходящие при передаче из-за влияния помех. Наиболее простым корректирующим кодом является код, обнаруживающий ошибки. Идея возможности обнаружения ошибок состоит в том, что для передачи используются не все кодовые комбинации $N = 2^n$ , а лишь некоторая часть их $N_0 < N$ . Используемые в данном коде комбинации называют разрешенными, а оставшиеся $N - N_0$ неиспользуемых комбинаций - запрещенными. Если в результате ошибок разрешенная комбинация превращается в одну из запрещенных, то тем самым и обнаруживается наличие ошибок. Поскольку для получения заданного числа $N_0$ разрешенных комбинаций требуется $K$ двоичных разрядов, где $K = \log_2 N_0$, то остав-

шиеся $n-K$ разрядов характеризуют абсолютную избыточность данного кода.

На практике удобно пользоваться таким кодом, у которого информационные и избыточные разряды разделены. Такой код еще называют систематическим, или разделимым. Далее, относительная избыточность или скорость передачи определяется как отношение числа избыточных разрядов кода к числу информационных разрядов:

$$R = \frac{n-K}{K}.$$

Пример. Наиболее распространенным кодом для обнаружения ошибок является код с проверкой на четность. Допустим, что необходимо передать кодовое слово IIIO. С помощью схемы проверки на четность подсчитываем признаки "Чет" или "Нечет". Условно принимаем, что если в кодовом слове число единиц нечетно, признак "Нечет" равен единице. Таким образом в нашем случае передается избыточное кодовое слово IIIOI. На приемной стороне это слово принимается и проверяется на четность числа единиц в слове. Если ошибки при передаче не было, то принятый контрольный разряд и вновь вычисленный совпадут. Если же в процессе передачи один из разрядов изменит свое значение на обратное, то совпадения контрольных разрядов не произойдет. Нетрудно заметить, что ошибка будет обнаружена также, если произойдет искажение контрольного разряда.

Важным параметром кода является кодовое расстояние. Очевидно, что код тем лучше приспособлен к исправлению ошибок, чем больше отличаются друг от друга кодовые слова. По Хэммингу кодовым расстоянием $d$ между двумя кодовыми словами называется число несовпадающих позиций этих слов. Например, складывая по модулю два слова II00I0 и 00I0I0, получим:

II00I0
00I0I0
III000,

т. е. в нашем примере $d = 3$.

На практике более широкое распространение получило понятие минимального кодового расстояния $d_0$. Суть дела в том, что для гарантии корректирующей способности кода $t$, необходимо, чтобы между различными кодовыми словами сохранялось минимальное расстояние между любыми двумя кодовыми комбинациями. В теории кодирования имеется следующее важное утверждение: код способен исправлять любые комбинации из $t$ (и меньшего числа) ошибок, тогда и только тогда, когда его кодовое расстояние больше 2 или $d_0 = 2t + 1$. Однако, если нам потребуется построить код только для обнаружения ошибок, то достаточно, чтобы выполнялось неравенство:



$$d_0 \geq t + 1.$$

Интересно отметить, что в натуральном двоичном коде $d_0 = 1$.

Линейные коды. Эти коды образуют наиболее обширный с практической точки зрения класс кодов. Такое название линейные коды получили потому, что как информационные, так и проверочные символы представляют собой различные линейные комбинации информационных символов, и поэтому способ декодирования основан на проверке линейных соотношений между символами. Чаще всего при построении двоичных кодов в качестве линейной операции выбирают сложение по модулю два. Тогда линейные проверки сводятся к обобщенным проверкам на четность. Линейные коды называют еще групповыми кодами, поскольку они образуют алгебраическую группу по отношению к операции посимвольного сложения по модулю два и обозначают как ($n$, к) коды, где $n$ - общее число разрядов в кодовом слове и к - число информационных разрядов кода.

Понятие синдрома кодового слова. Слово "Синдром" означает обычно совокупность признаков, характерных для того или иного явления. Такой же примерный смысл имеет понятие "Синдром" и в теории кодирования. Зная синдром, можно определить характер ошибок. Синдромом кодового слова А называется кодовое слово, определяемое из равенства

$$C = AH^T, \qquad (12)$$

где $H^T$ - матрица, транспонированная к проверочной матрице Н. Как мы уже отметили, код с проверкой на четность с обнаружением одной ошибки имеет одну проверку на четность по всем $K$ символам. Поскольку в таком коде в проверках участвуют все $K$ символов, то способ получения проверочного символа можно записать в виде тривиальной матрицы

$$H = IIII \ldots I. \qquad (13).$$

Характерной особенностью матрицы (13) является следующее:
- она описывает схему проверочных соотношений как для кодирующей, так и для декодирующей схемы;
- число столбцов матрицы равно числу информационных символов;
- число строк матрицы равно числу проверочных символов.

Пример. Известное нам кодовое слово А = IIIOI с одним проверочным символом справа, и проверяется на приемной стороне на четность в соответствии с соотношением (12)

$$C = IIIOI \begin{bmatrix} I \\ I \\ I \\ I \\ I \end{bmatrix} = I \begin{bmatrix} I \\ I \\ I \\ I \end{bmatrix} + I \begin{bmatrix} I \\ I \\ I \\ I \end{bmatrix} + I \begin{bmatrix} I \\ I \\ I \\ I \end{bmatrix} + O \begin{bmatrix} I \\ I \\ I \\ I \end{bmatrix} + I \begin{bmatrix} I \\ I \\ I \\ I \end{bmatrix} = 0.$$

Допустим, что символ А передан с ошибкой, например, вместо единицы в крайней позиции слева передан нуль. Тогда, как нетрудно проверить, значение синдрома С не будет равно нулю. И вообще, в синдромном кодировании справедливы следующие утверждения. Пусть вектор А не является кодовым, тогда этот вектор (КОДОВОЕ СЛОВО) обязательно содержит ошибочные символы. Вектор А можно представить тогда в виде суммы посланного кодового вектора В (который пока неопределен) и вектора ошибки е :

$$A = B + e . \qquad (I4)$$

Вектор е содержит ненулевые символы в тех позициях , в которых вектор А содержит искаженные символы. Умножив обе части равенства (I4) на $H^T$ , получим:

$$AH^T = (B + e) H = BH^T + eH^T = eH^T = C \qquad (I5).$$

Из равенства (I5) следует важное правило, широко используемое в практике кодирования: синдром принятого вектора А и вектора ошибки совпадают. Это обстоятельство позволяет исправлять ошибки в принятом кодовом векторе . Следует отметить, что в литературе по теории кодирования наряду с выражением (I2) используется также эквивалентное ему следующее выражение для вычисления синдрома:

$$C = \bar{A}H^T. \qquad (I6)$$

Соотношения (I2) и (I6) примечательны тем, что с их помощью можно находить также те кодовые слова, которые принадлежат выбранному коду.

Существует ряд примечательных кодов, которые можно построить простым способом с помощью проверочных матриц. В качестве примера рассмотрим код Хэмминга.

Код Хемминга. Хотя порядок столбцов безразличен, но чаще всего их упорядочивают так , чтобы содержимое каждого столбца являлось двоичной записью его номера. Проверочная матрица кода Хэмминга длины I5 ($m = 4$) имеет вид:

$$H = \begin{vmatrix} 000000011111111 \\ 000111100001111 \\ 011001100110011 \\ 101010101010101 \end{vmatrix} . \qquad (I7)$$

Прежде чем рассматривать алгоритм исправления одиночных ошибок (код Хэмминга имеет $d = 3$), рассмотрим процедуру кодирования, которая выполняется на стороне передатчика. Для этого необходимо знать правило вычисления проверочных соотношений. Простой перестановкой столбцов, содержащих одну единицу , матрицу (I7) можно привести к виду



$$H = \begin{vmatrix} 000001111111000 \\ 011100011110100 \\ 101101100110010 \\ 110110101010001 \end{vmatrix} = EI, \qquad (I8)$$

где I - единичная матрица порядка 4.

Данная матрица H соответствует коду Хэмминга (I5, II) с минимальным расстоянием $d = 3$. Использование такого кода позволяет исправить любую одиночную ошибку или обнаружить произвольную ошибку кратности 2. Считаем, что нумерация информационных и контрольных разрядов разделимого кода производится слева направо. Тогда в соответствии с матрицей (I8) получается система линейных уравнений, с помощью которых вычисляются проверочные (контрольные разряды)

$$\begin{aligned} c_1 &= к_5 + к_6 + к_7 + к_8 + к_9 + к_{10} + к_{11} \\ c_2 &= к_2 + к_3 + к_4 + к_8 + к_9 + к_{10} + к_{11} \\ c_3 &= к_1 + к_3 + к_4 + к_6 + к_7 + к_{10} + к_{11} \\ c_4 &= к_1 + к_2 + к_4 + к_5 + к_7 + к_9 + к_{11}, \end{aligned} \qquad (I9)$$

где $c_i$ - контрольные разряды, $к_i$ - информационные разряды. Рассмотрим пример. Допустим, что необходимо передать следующие информационные разряды II000I0000I . Тогда из соотношений (I9) имеем: $c_1 = 0$, $c_2 = 0$, $c_3 = I$ и $c_4 = I$. В итоге имеем закодированное слово II000I0000I00II. Нетрудно заметить, что если в процессе передачи кодовое слово передается верно, то контрольные соотношения (I9) будут выполненными, и вычисленный синдром (корректор) равен нулю : $p_1 = 0 + I + 0 + 0 + 0 + 0 + I + 0 = 0$, так как $c_1 = 0$. Далее $p_2 = I + 0 + 0 + 0 + 0 + 0 + I + 0$, $p_3 = I + 0 + 0 + I + 0 + 0 + I + I = 0$ и $p_4 = 0$. Если же при передаче кодового слова возникнет ошибка, то окажутся невыполненными контрольные соотношения (I9). Например, если ошибка возникла во втором информационном символе, то окажутся невыполненными второе и 4-е уравнения, и синдром будет равен 0I0I, совпадая со вторым столбцом матрицы (I8). И, вообще: местоположение столбца матрицы (I8), совпадающего с вычисленным синдромом, в случае появления одиночной ошибки, указывает место ошибки. Если же возникнет ошибка кратности 2, то полученное значение синдрома, как это нетрудно проверить, также будет совпадать с одним из столбцов матрицы (I8), так как сумма по модулю два всевозможных столбцов этой матрицы окажется равной одному из столбцов. Широко известен также модифицированный код Хэмминга с минимальным расстоянием $d = 4$ . Такой код можно получить из обычного кода Хэмминга добавлением одного контрольного соотношения, представляющего собой результат суммирования по модулю два всех разрядов кодового слова. С помощью такого кода можно исправлять одиночные и об-

наруживать кратные ошибки (23). В качестве аппаратных средств для вычисления обобщенных проверок на четность (для вычисления синдрома) используются известные нам сумматоры по модулю два или много – входовые схемы проверки на четность.

Изучая теорию и практику корректирующих кодов, автор пришел к выводу о том, что декодирующие устройства можно использовать для параллельной регистрации множественности событий в годоскопических системах (4).

Циклические коды. Циклические коды являются наиболее популярными, так как наряду с важными с практической точки зрения свойствами они отличаются довольно простой техникой кодирования и декодирования. Ряд циклических кодов имеет алгебраическую структуру, например, элементы поля Галуа $GF(2^m)$. Основное свойство циклических кодов, определяющих их название, состоит в том, что если кодовый вектор $A = (a_0, a_1, a_2 \ldots, a_{n-1})$ принадлежит коду, то и вектор A, полученный из A циклической перестановкой составляющих, т. е. $A = (a_1, a_2, a_3 \ldots a_0)$, также принадлежит коду. Известно несколько способов построения циклических кодов (10, 12, 20). Циклический код полностью задается порождающим многочленом $g(X)$, на который делится многочлен $X^n + 1$. Если $g(X)$ - многочлен степени $r$, то размерность кода равна $k = n - r$.

Пример. Дан многочлен $X^7 + 1 = (X+1)(X^3 + X + 1)(X^3 + X^2 + 1)$ над полем $GF(2)$. Многочлен $g(X) = X^3 + X + 1$ порождает циклический (7, 4) - код. Элементы

$$\begin{aligned} g(X) &= (1\ 0\ 1\ 1\ 0\ 0\ 0) \\ Xg(X) &= (0\ 1\ 0\ 1\ 1\ 0\ 0) \\ X^2g(X) &= (0\ 0\ 1\ 0\ 1\ 1\ 0) \\ X^3g(X) &= (0\ 0\ 0\ 1\ 0\ 1\ 1) \end{aligned} \qquad G = \begin{vmatrix} 1\ 0\ 1\ 1\ 0\ 0\ 0 \\ 0\ 1\ 0\ 1\ 1\ 0\ 0 \\ 0\ 0\ 1\ 0\ 1\ 1\ 0 \\ 0\ 0\ 0\ 1\ 0\ 1\ 1 \end{vmatrix}$$

можно выбрать в качестве базисных векторов, и, следовательно, матрицу $G$ в качестве порождающей матрицы для этого кода.

Теперь осталось найти проверочную матрицу такого кода. Из теории известно, что проверочная матрица может быть получена из соотношения $h(X) = (X^n + 1)/(X^3 + X^2 + 1) = X^4 + X^3 + X^2 + 1$. Тогда имеем:

$$\begin{aligned} h(X) &= (1\ 0\ 1\ 1\ 1\ 0\ 0) \\ Xh(X) &= (0\ 1\ 0\ 1\ 1\ 1\ 0) \\ X^2h(X) &= (0\ 0\ 1\ 0\ 1\ 1\ 1). \end{aligned} \qquad (20)$$

Как это показано в работе (11), проверочную матрицу для данного кода можно получить, если компоненты векторов в соотношениях (20) переписать в обратном порядке. После таких преобразований получаем матрицу :



$$H = \begin{vmatrix} 0\ 0\ 1\ 1\ 1\ 0\ 1 \\ 0\ 1\ 1\ 1\ 0\ 1\ 0 \\ 1\ 1\ 1\ 0\ 1\ 0\ 0 \end{vmatrix}.$$

Можно проверить, что $G H^T = 0$. Практически это значит, что мы получили циклический код Хэмминга, содержащий 4 информационных и 3 проверочных разряда, так как матрица H содержит в качестве столбцов все различные ненулевые последовательности длины 3.

Рассмотрим теперь общий вид порождающей и проверочной матрицы. Для этого представим многочлен $g(X)$ в виде

$$g(X) = g_0 + g_1X + \ldots + g_rX^r. \qquad (21)$$

Степень многочлена (21) равна $k = n - r$.

Рассмотрим следующие многочлены:

$$g(X),\ Xg(X),\ X^2g(X),\ \ldots\ X^{k-1}g(X). \qquad (22)$$

Все они являются кодовыми и их степень не превосходит $n - 1$. Рассматриваемые как векторы, они образуют линейно независимую систему, и всякий кодовый вектор является их линейной комбинацией. Поэтому матрица, составленная из векторов (22), является порождающей матрицей циклического кода. Порядок ее равен $k \times n$, и она имеет следующий вид:

$$G = \begin{vmatrix} g_0 & g_1 & \ldots & g_r & 0 & \ldots & 0 \\ 0 & g_0 & g_1 & & g_r & \ldots & 0 \\ \ldots & & & & & & \\ 0 & \ldots & 0 & g_0 & g_1 & \ldots & g_r \end{vmatrix} = \begin{vmatrix} g(X) \\ Xg(X) \\ \ldots \\ X^{k-1}g(X) \end{vmatrix}.$$

Аналогично проверочная матрица циклического кода порядка $r \times n$ имеет вид:

$$H = \begin{vmatrix} 0 \ldots & & 0\ h_k \ldots h_1 h_0 \\ 0 \ldots & & h_k \ldots h_0\ 0 \\ \ldots & & \ldots \\ h_k & \ldots & h_0\ 0\ 0 \end{vmatrix} = \begin{vmatrix} h(X) \\ Xh(X) \\ \ldots \\ X^{r-1}h(X) \end{vmatrix}.$$

Первая строка этой матрицы составлена из коэффициентов многочлена $h(X)$, записанных в обратном порядке (т. е. в порядке убывания степеней). Последующие строки составлены аналогичным образом из коэффициентов многочленов $Xh(X), \ldots, X^{r-1}h(X)$.

На конкретных примерах рассмотрим, каким образом осуществляется процесс кодирования и декодирования циклических кодов.

Цикличность перестановок, определяющих строение рассматриваемых кодов, лежит в основе техники кодирования и декодирования. Хотя в основе этой техники лежат сдвиговые регистры с логической обратной

связью, в последнее время в связи с резким уменьшением стоимости интегральных микросхем, появилась возможность выполнять эти процедуры и параллельным способом.

Работу кодирующего устройства последовательного типа рассмотрим на примере того же циклического кода, порождаемого полиномом $g(X) = X^3 + X + 1$. Соответствующая данному полиному схема кодирующего устройства приведена на рис.24. Пусть на вход подается последовательность информационных символов 1001. В нижеследующей таблице цикл за циклом показано формирование контрольных символов в регистре. В исходном состоянии в регистре записаны нули. В левом столбце приведены информационные символы к. В правом столбце - символы с, вытесняемые при каждом цикле из регистра. В цепь обратной связи поступает сумма по модулю два: $p = к + с$:

| к | 0 | 0 | 0 | с |
|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 |

Следующие три цикла выводят контрольные символы из регистра, образуя кодовый вектор 1001110. В качестве декодирующего устройства может быть использована схема, изображенная на рис. 25. Действие этой схемы сводится к тому, что принятый вектор вводится в нее последовательными циклами. Другими словами, с помощью такой схемы вычисляется синдром. Если ошибки нет, то содержимое регистра будет равно нулю. Если же синдром не равен нулю, то это указывает на наличие ошибки. Положение ошибочного символа определяется следующим образом. При отключении входа содержимое регистра последовательно сдвигается, и номер цикла сдвига, на котором в регистре появляется код из единицы в первой ячейке и нулей во всех остальных, и есть номер ошибочного символа.

Пример. Будем сначала вводить в декодер кодовый вектор без ошибки, например, 100 1011, содержащий 4 информационных и три контрольных разряда. Ключ К замкнут. Тогда получим следующее:

| к | 0 | 0 | 0 | с |
|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |

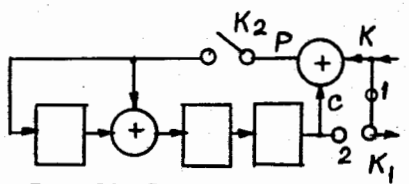


Рис. 24. Схема кодирующего устройства.



Так как синдром равен нулю, то считаем, что имела место безошибочная передача. Пусть теперь принята последовательность 1001010, т.е. первый информационный символ принят с ошибкой. Вычисляем синдром:

| к | 0 | 0 | 0 | с |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 |

Рис. 25. Схема декодирующего устройства.

После 7-го цикла ключ К размыкается, и следующие три цикла сдвигают образовавшееся сочетание 101. Как нетрудно проверить, требуемое сочетание 100 образуется в регистре сразу же после первого такта. Это значит, что ошибка содержится в первом символе принятой последовательности.

Из рис. 24 и 25 видно, что в качестве устройств для кодирования и декодирования используется счетчик в поле Галуа $GF(2^3)$. Назначение ключей $К_1$ и $К_2$ на рис. 24 следующее. Вначале ключ $К_1$ находится в положении 1, а ключ $К_2$ замкнут. Информационные символы, подаваемые на вход, во-первых, непосредственно передаются в линию, а во-вторых, поступают в кодирующее устройство, где за $k$ шагов образуется $n-k$ контрольных символов. После этого ключ переключается в положение 2, соединяя регистр с выходом, а ключ $К_2$ в цепи обратной связи размыкается. Затем регистр сдвигается еще $n-k$ циклов, и контрольные символы поступают на выход.

Коды, исправляющие пакеты (кластеры) ошибок. До сих пор мы рассматривали коды, исправляющие одиночные ошибки. Но существует класс специальных циклических кодов, которые применяются для исправления пакетов ошибок или кластеров. Такие ошибки возникают, например, в накопителях на магнитных лентах и магнитных дисках. Пакет ошибок длиной $b$ определяется вектором ошибки $e$, в котором все единицы заключены в последовательности $b$ символов при условии, что крайние символы этой последовательности - единицы. Так, пакеты ошибок длиной 5 могут выглядеть следующим образом ($n = 12$):

000111110000, 000100010000 и т. п.

Широкое применение получили коды Файра, которые удобнее всего определять с помощью порождающего многочлена

$$g(X) = p(X)(X^c + 1),$$

где $p(X)$ - неприводимый многочлен степени $m$ над полем $GF(2^m)$.

Код Файра имеет следующие свойства: значимость кода $n$ есть общее наименьшее кратное показателя $a$ и порядка корней полинома $p(X)$, равного $e = 2^m - I$; число проверочных символов $n - k = c + m$, число информационных символов $k = n - c - m$.

Используя эти коды, можно исправить любую одиночную пачку ошибок длины $b_c$ или меньше и одновременно обнаружить любую пачку ошибок длины $b_d \geq b_c$ или меньше, если $c \geq b_c + b_d - I$ и $m \geq b_c$. Если применять эти коды только для обнаружения ошибок, то можно обнаружить любую комбинацию из двух пачек ошибок, длина наименьшей из которых не превосходит $m$, а сумма длин которых не превосходит $e + I$ так же, как и любую одиночную пачку ошибок длины, не превосходящей $c + m$ - числа проверочных символов.

Пример. Определим код Файра из следующего соотношения (24):

$$g(X) = (X^{11} + X^2 + I)(X^{68} + I),$$

где $(X^{11} + X^2 + I)$ представляет собой неприводимый полином (таблицы неприводимых полиномов степени I6 и меньше даны в (II). В соответствии с теорией свойства такого кода следующие:

- поскольку $e = 2^m - I = 2^{11} - I = 2047$, то $n$ = н.о.к. (2047, 68) = I39I96;
- число проверочных символов: $c + m = 68 + II = 79$;
- число информационных символов: $k = n - c - m = I39II7$;
- эффективность кода, равная отношению числу информационных символов к общему числу символов в коде, равна 99, 94 %, т. е. очень высока.

Следует отметить, что по структуре образующего полинома можно определить схему кодирующего и декодирующего устройства следующим образом:

$$g(X) = X^{79} + X^{70} + X^{68} + X^{11} + X^2 + I \quad \text{или то же самое}$$

$$X^{79} = X^{70} + X^{68} + X^{11} + X^2 + I.$$

Таким образом число разрядов сдвигового регистра в нашем примере равно 79, а обратная связь подается на первый, второй, одиннадцатый, шестьдесят восьмой и семьдесят восьмой разряды.

Нетрудно заметить аналогию, которая имеется между кластерами, возникающими в устройствах связи и вычислительной техники и одновременным срабатыванием соседних датчиков в многоканальных детекторах заряженных частиц. Поэтому кодирующие и декодирующие устройства кода Файра могут быть использованы для эффективной регистрации информации в спектрометрах физики высоких энергий в соответствии с изложенным выше методом синдромного кодирования. В работе (25) даны таблицы порождающих полиномов кодов Файра.



БЧХ- коды. Наиболее важным классом циклических кодов, исправляющих многократные независимые ошибки, являются коды Боуза - Чоудхури - Хоквингема (БЧХ-коды). Эти коды удобно задаются с помощью корней порождающего полинома $g(X)$. Этот полином имеет в качестве корней элементы поля Галуа $GF(2^m)$. Четные степени элемента $a$ могут быть отброшены, так что последовательность корней полинома $g(X)$ имеет вид:

$$a, a^3, a^5, \ldots a^{2t-1}.$$

Оказывается, что полином, удовлетворяющий этому условию, порождает код, исправляющий все ошибки кратности $\leq t$. Для построения полинома $g(X)$ следует образовать минимальные полиномы (или минимальные функции) $m_i(X)$ $(i = I, 3, 5, \ldots, 2t - I)$. В работе (26) дается простой способ получения минимальных полиномов, суть которого мы рассмотрим на примере поля $GF(2^4)$ и для случая, когда необходимо построить БЧХ-код с $n = I5$ и $t = 3$. Еще раньше мы установили, что элемент $a$ является корнем полинома $X^4 + X + I$ и поэтому является минимальным. Итак, имеем $m_1(X) = X^4 + X + I$.

По определению, минимальным полиномом $m_A(X)$ произвольного ненулевого элемента A в поле $GF(2^m)$ является единственный полином с двоичными коэффициентами и наименьшей степени, корнем которого является элемент A. Далее, этот полином имеет в качестве корней все $K$ различных элементов в виде $A, A^2, A^4, A^8, \ldots, A^{2^{K-1}}$, где $K$ является наименьшим числом, таким, что выполняется равенство

$$A^{2^K} = A.$$

Это значит, что минимальный многочлен может быть разложен на множители

$$m_A(X) = (X + A)(X + A^2)(X + A^4) \ldots (X + A^{2^{K-1}}).$$

Причем здесь рассматриваются два случая кода $K < m$ и $m = K$.

Пример I, $K < m$. Найти минимальный полином элемента $A = a^5$ в поле $GF(2^4)$. Решение: различными корнями $m_A(X)$ являются $a^5$ и $a^{10}$, поскольку $a^{20} = a^5$. Поэтому $m_A(X) = (X + a^5)(X + a^{10})$. Далее $m_A(a) = (a + a^5)(a + a^{10}) = a^2 a^8 = a^{10} = III0$ или то же самое, что

$$m_5(X) = X^2 + X + I.$$

Пример 2, $K =$ : Найти минимальный полином элемента $a^3$.

Решение: различными корнями этого полинома будут $a^3$, $a^6$, $a^{12}$ и $a^{24} = a^9$, так как $a^{18} = a^3$, то

$$m_3(a^3) = (a + a^3)(a + a^6)(a + a^{12})(a + a^{9}) = a^6 = 0011.$$

Поскольку $K = m$, то по правилу, приведенному в работе (26), необходимо к $m_A(a)$ добавить по модулю два $m_1(a) = I + a + a^4$, т. е. II00I. В результате получим IIII, и, окончательно

$$m_3(X) = X^4 + X^3 + X^2 + X + I.$$

В конечном итоге получаем порождающий полином БЧХ-кода, с параметрами : корректирующая способность $t = 3$, $n = 15$, число проверочных символов $r = 12$ и число информационных символов $k = 15 - 12 = 3$. Декодирование БЧХ-кодов мы здесь не рассматриваем. Этот непростой вопрос достаточно подробно изложен в книгах по теории кодирования (10, 11,20) . Применение теории БЧХ-кодов для создания быстродействующих устройств отбора физических событий описано в работе (14). Как показывает практика, наиболее эффективными устройствами, применяемыми для декодирования БЧХ-кодов, являются ППЗУ.

## Приложение

### Комплекс программ для автоматизации логического проектирования устройств сжатия информации, разрабатываемых на базе алгебраической теории кодирования

В предыдущих разделах нами были рассмотрены вопросы применения теории кодирования для создания устройств **сжатия данных**, регистрируемых в годоскопических системах. Было показано, что, используя эту теорию, можно строить эффективные параллельные шифраторы и быстродействующие устройства для комбинаторного отбора событий. Ниже приводится краткое описание комплекса программ для автоматизации **логического** проектирования и моделирования подобных устройств. Большая часть программ **написана** на языках аналитических преобразований **Schoonship и Reduce.**

Некоторые программы составлены на автокоде для ЭВМ ЕС-1010 и на языке *PL/1.*

Программа нахождения элементов поля Галуа. Элементы поля Галуа $GF(2^m)$ - многочлены переменной X, принимающей значения 0 или 1 с коэффициентами, равными 0 или 1, для которых операции сложения, вычитания, умножения и деления определены специальным образом. Правила выполнения большинства из этих операций были рассмотрены выше.

Программа TABL для заданного неприводимого многочлена степени $m$ вычисляет $2^m - 1$ элементов поля Галуа $GF(2^m)$. Например, при $m = 3$ для неприводимого многочлена $X^3 + X + 1$ элементами поля $GF(2^3)$ являются многочлены $a^0 = 1$, $a^1 = X$, $a^2 = X$, $a^3 = X + 1$, $a^4 = X^2 + X$, $a^5 = X^2 + X + 1$, $a^6 = X^2 + 1$, $a^7 = 0 = 1$.

Программа написана на языке *Schoonschip.*

Программа составления матрицы проверочных соотношений

Матрица $H^T$ имеет вид



$$H^T = \begin{vmatrix} 1 & 1 & 1 \\ a & a^3 & \dots a^{2t-1} \\ \vdots & \vdots & \vdots \\ a^{n-1} & a^{3(n-1)} & a^{(2t-1)(n-1)} \end{vmatrix}$$

Число строк этой матрицы равно числу разрядов входной информации (например, числу проволочек в многопроволочной пропорциональной камере) , число столбцов - числу разрядов выходной сжатой информации (синдром кода). Программа MATR, написанная на языке *Reduce*, вычисляет коэффициенты полиномов, являющихся элементами матрицы $H^T$.

Программа моделирования работы устройства сжатия информации. Приведенная выше матрица проверочных соотношений (матрица связей) $H^T$ является , по существу, принципиальной схемой устройства сжатия информации. Для моделирования его работы используется программа RACH, написанная на автокоде для ЭВМ ЕС-1010.

В основу программы положен алгоритм У. Питерсона [11]. Координаты сработавших проволочек пропорциональной камеры или сцинтилляторов, от которых поступили сигналы, определяются из уравнения

$$X^3 + \sigma_1 X + \sigma_2 X + \sigma_3 = 0. \qquad (\Pi 1)$$

Прежде всего вычисляются $\sigma_1$, $\sigma_2$ , $\sigma_3$ по следующим формулам:

$$\sigma_1 = S_1 \; ; \quad \sigma_2 = \frac{S_1^2 S_3 + S_5}{S_1^3 + S_3} \; ; \quad \sigma_3 = \frac{S_1 S_5 + S_3^2 + S_1^3 S_3 + S_1^6}{S_1^3 + S_3} \qquad (\Pi 2)$$

$S_1$, $S_3$, $S_5$, 2 формируются в блоке сжатия информации, а в память ЭВМ коды $S_1$, $S_3$, $S_5$ поступают в виде $n$- разрядных слов.

Программа вычисляет элементарные симметрические функции $\sigma_1, \sigma_2, \sigma_3$ по формулам П2, затем, подставив их значения в выражение П1, поступает следующим образом.

Берем массив элементов поля Галуа $(2^m)$. Поочередно подставляем каждый элемент массива в уравнение П1. Те элементы, которые удовлетворяют этому уравнению, и соответствуют номерам сработавших проволочек (сцинтилляторов).

Программа вычисления детерминанта матрицы. Программа DET, написанная на языке *Reduce*, вычисляет коэффициенты полинома, являющегося детерминантом матрицы

$$L_t = \begin{vmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ S_2 & S_1 & 1 & 0 & \cdots & 0 \\ S_4 & S_3 & S_2 & S_1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ S_{2t-4} & S_{2t-5} & S_{2t-6} & S_{2t-7} & \cdots & S_{2t-3} \\ S_{2t-2} & S_{2t-3} & S_{2t-4} & S_{2t-5} & \cdots & S_{2t-1} \end{vmatrix}$$

Порядок матрицы - от 3 ДО 10 вычислен на ЭВМ.

Программа вычисления переключательных функций. Программа КОЕР, написанная на языке *Reduce*, вычисляет переключательную функцию, представленную в виде полинома (8) (см. стр. 25). Она включает также подпрограмму для вычисления коэффициентов A(K).

Для вычисления переключательных функций, представляемых полиномами, имеющих степени более, чем 3, была составлена программа на языке PL/I, которая занимает 130К оперативной памяти. На данном этапе возможности программы таковы, что в процессе вычисления при каждом базисном элементе суммарное количество символов в сомножителях должно быть одинаковым (X считаем за один символ) и не должно превышать 250. Вычисления выполняются итеративно:

При $m = 4$ И $m = 5$ время, затрачиваемое центральным процессором, составляет 1 и 5 мин соответственно.

Литература


1. Лабунец В. Г. Алгебраическая теория сигналов и систем. Из-во Красноярского университета, Красноярск, 1984.
2. Блейхат Р. Э. Алгебраические поля, обработка сигналов, контроль ошибок. ТИИЭР, 1985, том 73, № 5, стр. 30-53.
3. Аршинов М. Н., Садовский Л. Е. Коды и математика. Библиотека "Квант", вып. 30, М. "Наука", 1983.
4. Никитюк Н. М. Вопросы оптимального кодирования данных в годоскопических системах. ПТЭ, 1983, № 3, стр. 74.
5. Александров И. Н., Гайдамака Р. И., Никитюк Н. М., Шириков В. П. Расчет переключательных функций, представленных элементами поля Галуа $GF(2^m)$. Препринт ОИЯИ, Р10-84-865, Дубна, 1984.
6. Никитюк Н. М. Новый способ построения универсального логического модуля. Препринт ОИЯИ, Р11-85-365.





7. Гордон Г., Надиг Н. Локализация неисправностей в микропроцессорных системах при помощи шестнадцатиричных ключевых кодов. Электроника , 1977, том 50, № 5 , стр. 24-33.
8. Смирнов Н. И., Стручков А. А., Судовцев В. А. Диагностика неисправностей в цифровой радиоаппаратуре на БИС. Зарубежная радиоэлектроника, 1979 , № 1, стр. 53-60.
9. **Savir j. Syndrome-testable design of combinational circuits. IEEE Transaction on computers, 1 980, vol. c-29, No, 6.**
10. Берлекэмп Э. Алгебраическая теория кодирования. "Мир", М. 1971.
11. Питерсон У., Уэлдон Э. Коды, исправляющие ошибки. "Мир", М. 1976.
12. Колесник В. Д., Мирончиков Е. Т. Декодирование циклических кодов. "Связь", М. 1968.
13. Гайдамака Р. И., Никитюк Н. М., Шириков В. П. Комплекс программ для автоматизации логического проектирования устройств сжатия информации, разрабатываемых на базе алгебраической теории кодирования. Препринт ОИЯИ, Р-84-84, Дубна, 1984.
14. Никитюк Н. М. Метод синдромного кодирования и его применение для быстрого аппаратного отбора событий на основе процессоров, оперирующих в поле Галуа $GF(2^m)$. Препринт ОИЯИ, Р11-80-484, Дубна, 1980.
15. **Bartee T.C. , SChneider P.I. Computation with fields. Information and Control, 1963, vol. 6, No. 1.**
16. Гайдамака Р. И., Калинников В. А., Никитюк Н. М., Шириков В. П. Новый способ построения мажоритарных схем совпадений. Препринт ОИЯИ, Р13-82-628, Дубна, 1982.
17. **Schiavita E, Soso F. Proposal for a new organization of desision on electronics in multicounter experiments. Nucl. Instr, and Meth.,1968, vol. 60, No. 1, p. 36.**
18. **Colla G, Marconi C., Pilastrini R., Volta A. A programmable fast logic unit. Nucl. Instrum. and Meth., 1980, vol. 196, No3.**
19. Толстяков В. С., Номоконов В. Н., Карповский М. Г. и др. Обнаружение и исправление ошибок в дискретных устройствах. " Сов. радио" 1972.
20. Бородин Л. Ф. Введение в теорию помехоустойчивого кодирования. " Сов. радио", М. , 1968.
21. Харкевич А. А. Борьба с помехами. "Наука", М. , 1965.
22. Хэмминг Р. В. Теория кодирования и теория информации. "Радио и связь", М. , 1983.
23. Хетагуров Я. А., Ю. П. Руднев. Повышение надежности цифровых устройств методами избыточного кодирования. "Энергия", М. , 1974.
24. **Lignos D. Error detection and correction in mass storage equipment. Computer Design, 1972, vol. 11, No. 6, p. 71-75.**

25. **Wagner W.Best Fire codes with lenth up to 1200 bits. IEEE Trans. on Information Theory, 1970, vol IT- 16, p. 649-651.**

26. **Gordon J.A. Very simple method to find the minimum polinomial of a finite field. Electronic Letters, 1976, vol. 12, No. 25.**

27. Свердлик М.Б. Оптимальные дискретные сигналы. "Сов. Радио" М., 1975, 199 с.

28. Гердт В.А., Тарасов С.В., Ширков Д.В. Аналитические вычисления на ЭВМ в приложении к физике и математике. УФН, 1980, т. 130, с.113.

29. Компьютерная алгебра. Символьные и алгебраические вычисления. Ред. Бухбергер Б., Коллинз Дж., Лоос Р. Мир, М., 1986, 392 с.



Содержание

1. Введение ..... 1
2. Поле Галуа ..... 2
3. Примеры расширенных полей ..... 4
4. Основные алгебраические операции над элементами поля Галуа ..... 8
5. Совмещенные операции в поле Галуа ..... 11
6. Специализированный процессор в поле Галуа ..... 11
7. Расчет переключательных функций ..... 24
8. Быстрые программируемые логические модули ..... 30
9. Синдромное тестирование ..... 34
10. Элементы теории корректирующих кодов ..... 37
11. Приложение. Комплекс программ для автоматизации логического проектирования устройств сжатия информации, разрабатываемых на базе алгебраической теории кодирования 48
12. Литература ..... 51

Рукопись поступила в издательский отдел
10 июня 1987 года.

ПЕРЕЧЕНЬ

лекций, вышедших с 1974 в ОИЯИ

Фаустов Р.Н. Связанная система частиц в квантовой электродинамике. Вып.1. ОИЯИ, Дубна, 1974.

Синаев А.Н. Современные аппаратурные системы модульной структуры, используемые при создании измерительно-вычислительных комплексов /КАМАК, ВЕКТОР/. Вып.2. ОИЯИ, 8507, Дубна, 1975.

Волков Д.В. Кварки как следствие дуальности. Вып.3. ОИЯИ, Р2-8765, Дубна, 1975.

Пальчик М.Я., Фрадкин Е.С. Введение в теорию конформно-инвариантных квантовых полей. Вып.4. ОИЯИ, 2-8874, Дубна, 1975.

Замори З. Микропроцессоры. Вып.5. ОИЯИ, Р10-8852, Дубна, 1975.

Биленький С.М. Вопросы физики нейтрино высоких энергий. Вып.6. ОИЯИ, 2-9026, Дубна, 1975.

Малкин И.А., Манько В.И. Инварианты, когерентные состояния и динамические симметрии квантовых систем. Вып.7. ОИЯИ, Р2-9228, Дубна, 1975.

Волков М.К., Первушин В.Н. Квантовая теория поля с киральным лагранжианом и физика мезонов низких энергий. Вып.8. ОИЯИ, Р2-9390, Дубна, 1976.

Басиладзе С.Г. Интегральные схемы с эмиттерной связью и их применение в наносекундной ядерной электронике. Вып.9. ОИЯИ, 13-9744, Дубна, 1976.

Аникин С.А. и др. Перенормированные составные поля в квантовой теории поля. Вып.10. ОИЯИ, Р2-10528, Дубна, 1977.

Шляпников П.В. Множественные процессы и инклюзивные реакции. Вып.11. ОИЯИ, Р2-10681, Дубна, 1977.

Капусцик Э. Галилеева инвариантность в теории поля. Вып.12. ОИЯИ, Р2-10677, Дубна, 1977.

Бутцев В.С. Явление возбуждения высокоспиновых ядерных состояний и механизм поглощения отрицательных π-мезонов. Вып.13,ОИЯИ,Р15-10847, Дубна, 1977.

Валуев Б.Н. Применение алгебры Клиффорда к решению задачи Изинга - Онсагера. Вып.14. ОИЯИ, Р17-11020, Дубна, 1977.

Капусцик Э. Нестандартные алгебры квантово-механических наблюдаемых. Вып.15. ОИЯИ, Р4-11497, Дубна, 1978.



Блохинцев Д.И. Квантовая механика. Лекции по избранным вопросам. Вып.16. ОИЯИ, Р2-11728, Дубна, 1978.

Ширикова Н.Ю. Начинающим работать на ЭВМ CDC-6500. Вып.17. ОИЯИ, Р11-11739, Дубна, 1978.

Барбашов Б.М., Нестеренко В.В. Непрерывные симметрии в теории поля. Вып.18. ОИЯИ, Р2-12029, Дубна, 1978.

Некоторые проблемы физики высоких энергий /сборник/. Вып.19. ОИЯИ, Р2-12080, Дубна, 1978.

Басиладзе С.Г. Электронная регистрирующая аппаратура физического эксперимента. Вып.20. ОИЯИ, Р13-12151, Дубна, 1979.

Ефремов А.В., Радюшкин А.В. Партоны, жесткие процессы и квантовая хромодинамика. Вып.21. ОИЯИ, Р2-12803, Дубна, 1979.

Говорков А.Б. Введение в теорию кварков. Вып.22, ОИЯИ, Р2-12803, Дубна, 1979.

Говорков А.Б. Цветные кварки и глюоны. Вып.23. ОИЯИ, Р2-80-6, Дубна, 1980.

Исаев П.С. Глубоконеупругое рассеяние лептонов на нуклонах. Партонная модель нуклона. Вып.24. ОИЯИ, Р2-80-325, Дубна, 1980.

Казаков Д.И., Ширков Д.В. Суммирование асимптотических рядов в квантовой теории поля. Вып.25. ОИЯИ, Р2-80-462, Дубна, 1980.

Ососков Г.А. Применение методов распознавания образов в физике высоких энергий. Вып.26. ОИЯИ, Р10-83-187, Дубна, 1983.

Малышев В.А. Элементарное введение в математическую физику бесконечно-частичных систем. Вып.27. ОИЯИ, Р17-83-363, Дубна, 1983.

Савушкин Л.Н., Фоменко В.Н. Введение в мезонную теорию ядерных взаимодействий и ядерных систем. Вып.28. ОИЯИ, Р4-83-369, Дубна,1983.

Биленький С.М. Осцилляции нейтрино. Вып.29. ОИЯИ, Р2-83-441, Дубна, 1983.

Бужек В. Введение в метод стохастического квантования. Вып.30. ОИЯИ, Р2-84-419, Дубна, 1984.

Шумовский А.С., Юкалов В.И. Фазовые состояния и переходы. Вып.31. ОИЯИ, Р17-85-676, Дубна, 1985.

Владимиров А.А. Введение в квантовые интегрируемые системы. Метод R-матрицы. Вып.32. ОИЯИ, Р17-85-742, Дубна, 1985.

Осипов В.А., Федянин В.К. Полиацетилен и двумерные модели квантовой теории поля. Вып. 33. ОИЯИ, Р17-85-809, Дубна, 1985.

Шуян Ш. Стохастичность в динамических системах. Вып. 34. ОИЯИ, Р17-86-211, Дубна, 1986.

Ефремов А.В. Введение в квантовую хромодинамику. Вып. 35. ОИЯИ, Р2-86-212, Дубна, 1986.

Нестеренко В.В., Червяков А.М. Сингулярные лагранжианы. Классическая динамика и квантование. Вып. 36. ОИЯИ, Р2-86-323, Дубна, 1986.

Пепёлышев Ю.Н. Регистрация нейтронов /современное состояние и перспективы развития/ Вып. 37. ОИЯИ, Р13-86-719, Дубна, 1986.

Боголюбов Н.Н./мл./, Шумовский А.С. Светоизлучение. Вып. 38. ОИЯИ, Р17-87-176, Дубна, 1987.

Пушкаров Д.И. Дефектоны в кристаллах. /Метод квазичастиц в квантовой теории дефектов/. Вып. 39. ОИЯИ, Р17-87-177, Дубна, 1987.



## Требования, предъявляемые к серии брошюр "Лекции для молодых ученых ОИЯИ"

Серия брошюр "Лекции для молодых ученых ОИЯИ" издаётся с целью повышения научно-профессионального кругозора и уровня молодых ученых и специалистов ОИЯИ в актуальных областях исследований, ведущихся по тематике Института. Выпуски должны представлять собой законченные циклы лекций, прочитанные в ОИЯИ и ориентированные прежде всего на молодых сотрудников Института.

Лекции должны иметь характер учебного пособия, предназначенного для первого ознакомления с рассматриваемой проблемой, а также содержать обзор её современного состояния. Они должны быть снабжены подробным оглавлением и основной литературой. Большие параграфы рекомендуется разбивать на подпараграфы с вынесенными в оглавление подзаголовками.

Весь текст, включая отдельные главы и параграфы, следует печатать, заполняя каждую страницу целиком.

Рукопись должна быть напечатана на специальных бланках, предназначенных для прямого репродуцирования, которые можно получить в издательском отделе. Все формулы и схемы должны быть вписаны аккуратно и ясно тушью или чернилами черного цвета. Разметка формул не производится, их нумерация должна находиться в конце строки справа в круглых скобках. Текст лекций печатается на машинке с черной (не серой) лентой через I,5 интервала. Объём лекций не должен превышать I00 страниц машинописного текста.

Рукопись представляется в Редакционный совет серии брошюр "Лекции для молодых ученых ОИЯИ" Советом молодых ученых и специалистов ОИЯИ и Советами молодых ученых и специалистов лабораторий Института. Редакционный совет принимает окончательное решение о целесообразности ее публикации.

Редакционный совет