УДК 681.325.3:621.391.25:621.394.14

В.И. Кубицкий, И.А. Жуков

Метод непосредственного умножения элементов конечного поля $GF(2^m)$ с использованием логических функций

Предложен метод непосредственного умножения элементов конечного поля $GF(2^m)$ с использованием логических функций. Разработаны схемы устройств, реализующих данный метод, определены их сложности и дана сравнительная оценка этих сложностей и сложностей известных схем.

The method of the direct multiplication of Galois field elements $GF(2^m)$ using the logical operations is proposed. The device schemes supporting this method are elaborated, their difficulties are determined and the comparative evaluation of these difficulties and the difficulties of the known schemes are given.

Запропоновано метод безпосереднього множення елементів скінченного поля $GF(2^m)$ з застосуванням логічних функцій. Розроблено схеми пристроїв, які реалізують цей метод, визначено їх складності та подано порівняльну оцінку цих складностей і складностей відомих схем.

Введение. Вычисления в конечных полях используются во многих областях науки и техники. Операции в конечных полях лежат в основе алгоритмов кодирования и декодирования многих помехоустойчивых кодов. На основе этих операций могут быть реализованы алгоритмы цифровой обработки сигналов. Теория конечных полей используется в криптографии.

В виду важности конечных полей для развития современных радиоэлектронных систем самого разного назначения, необходимо совершенствовать ее технику. Учитывая возможности современной микроэлектроники, необходимо, чтобы перспективные методы выполнения операций в конечных полях были не только эффективными, но и технологичными, обязательно реализуемыми на базе больших интегральных схем (БИС) и должны обеспечивать однородность схем реализации.

Так как набор команд универсальных компьютеров не приспособлен для выполнения операций над элементами конечных полей, то для таких операций необходимо создавать дополнительные подпрограммы или специальные вычислительные устройства.

Вычисления в конечных полях могут быть реализованы различными методами в зависимости от способа представления элементов конечного поля.

Для каждой операции в $GF(p^m)$ удобен свой способ представления элементов конечного поля: для сложения элементов конечного поля – векторное и полиномиальное представление, для умножения и инвертирования в поле – степенное представление, при реализации операций по модулю некоторого многочлена – полиномиальное представление.

Для случая степенного представления элементов поля используются логарифмы Зеча. Для логарифмического представления используются таблицы логарифмов и антилогарифмов. Полиномиальное и векторное представление элементов поля позволяет выполнять операции над этими элементами на сдвиговых регистрах, в программируемых запоминающих устройствах (ПЗУ) или программируемых логических матрицах (ПЛМ), с использованием логических функций и других методов.

При разработке новых и усовершенствовании известных методов вычислений в конечных полях необходимо стремиться к улучшению таких параметров, как время вычислений и аппаратурная сложность средств, реализующих эти методы.

Состояние проблемы

Дадим общую характеристику некоторых наиболее часто используемых методов вычислений в конечных полях.

Вычисление на сдвиговых регистрах

Схемы, реализующие этот метод вычисления, находят наиболее широкое применение [1, 2]. Однако скорость вычисления для данного метода меньше, чем для метода вычисления с использованием логических функций. Но, как показывают исследования [3], при больших m в некоторых случаях (скажем, при $m \geq 9$ для

изменяемого неприводимого многочлена $p(x)$) этот метод может привести к экономии аппаратных средств.

Временная сложность схем умножения элементов (ССУЭ) конечного поля на сдвиговых регистрах равна [4]:

$$T^{\text{И}}_{\text{ССУЭ}} = (22m+15)t,$$

где t – время срабатывания элемента базисного набора (И, ИЛИ, НЕ).

Под *временной сложностью* (T) схемы понимается время, необходимое для реализации схемой заданной функции; при этом за единицу времени принимается время срабатывания элемента базисного набора (t).

При реализации вычислений на сдвиговых регистрах для большинства практически применяемых конечных полей время выполнения операций над элементами этих полей не обеспечивает возможности роста скоростей передачи информации. Схемы вычисления на сдвиговых регистрах не подходят для реализации в виде СБИС, не обладают модульностью и параллелизмом, а их структура неоднородна.

Вычисление с использованием логических функций

В этом случае вычисление может выполняться с помощью схем, построенных из логических элементов И и сумматоров по модулю 2. Характерным примером для этого метода есть метод, предложенный в [5] (метод Барти–Шнайдера), обеспечивающий высокую скорость вычислений. Однако, как показывает анализ, схемы устройств умножения элементов конечного поля, представленные в [5] и реализующие данный метод, имеют следующие недостатки:

- схемы пригодны только для фиксированного конечного поля;
- не разработаны схемы для вычисления элементов $\alpha_{kl}^{(i)}$ ($\alpha_{kl}^{(i)} \in GF(2)$ является i-й координатой произведения базисных элементов $\omega_k \cdot \omega_l$ поля $GF(2^m)$, $i = \overline{0, m-1}$, $1 \le k,l \le m$);
- величины c_i (c_i – результат умножения элементов конечного поля) вычисляются последовательно.

Время умножения на схеме, реализующей умножение по методу Барти–Шнайдера (схема БШУЭ) [4], составляет (при одновременном вычислении величин c_i и с учетом времени вычисления на схемах И величин ($a_j \cdot \alpha_{kl}^{(i)}$)):

$$T^{\text{И}}_{\text{БШУЭ}} = (6m-4)t.$$

Вычисления в ПЗУ

Для полей $GF(2^m)$ с небольшим числом элементов (для $m \le 6$) более эффективным (с учетом времени вычисления [6]) в сравнении с методом вычисления с использованием логических функций есть табличный метод, реализуемый в ПЛМ или ПЗУ. Очевидно, что в этом случае необходимо заранее вычислять результаты операций в поле $GF(2^m)$ перед тем, как занести их в ПЗУ или ПЛМ. Кроме этого, такая таблица будет отражать результаты операций для фиксированного конечного поля. Табличный метод также требует существенно большего количества оборудования, чем при реализации операций с использованием схем умножения и деления многочлена. Схема для умножения двух элементов поля $GF(2^m)$ будет иметь $2m$ входов и m выходов и может быть реализована на ПЗУ с размерностью информационного поля $2^{2m} \times m$ [6].

Использование таблиц логарифмов и антилогарифмов

При этом методе умножения (деления) элементов поля складываются (вычитаются) по модулю ($2^m - 1$) их логарифмы. Затем берутся антилогарифмы. Обратный переход необходим потому, что сложение элементов конечного поля выполняется проще, если эти элементы представлены в векторном виде или в виде многочлена. Таким образом, для умножения с помощью логарифмов необходимо иметь устройство сложения по модулю ($2^m - 1$) и две таблицы переходов от обычного представления к логарифмическому и обратно.

Метод, использующий таблицы логарифмов и антилогарифмов, хорош для $m \le 4$ поля $GF(2^m)$, но практически неприемлем при больших значениях m из-за необходимости иметь таблицы большого объема. Для конечного поля $GF(2^m)$

потребуется таблица логарифмов объемом памяти $2m(2^m-1)$ бит [7] и таблица антилогарифмов такого же объема [8]. Следует отметить, что нулевой элемент конечного поля не может быть представлен степенью примитивного элемента. Так что в таблице антилогарифмов не должно быть нулевого элемента поля, а в таблице логарифмов нулевому элементу поля не должно быть сопоставлено какое-либо число. Поэтому при работе с этими таблицами требуется специальная проверка на нулевой элемент поля в результате вычислений или в исходных данных. Это – недостаток. Недостаток также то, что таблицы формируются для заданного конечного поля и могут использоваться для вычислений только в пределах этого поля.

Анализ показывает, что перспективна реализация операций в конечных полях методами, использующими логические функции. Поэтому предложим метод умножения элементов конечного поля, основанный на использовании логических функций. Выведем математические выражения, разработаем алгоритмы и построим комбинационные схемы для реализации умножения элементов конечного поля $GF(2^m)$.

Решение

Математические выражения для умножения элементов конечного поля

Если элементы поля $GF(2^m)$ представлять в виде многочленов над полем $GF(2)$, степень которых меньше m, то умножение элементов a, $b \in GF(2^m)$ выполняется по правилу умножения представляющих эти элементы многочленов по модулю заданного неприводимого многочлена

$$p(x) = x^m + p_{m-1}x^{m-1} + \ldots + p_0 x^0 = x^m + h(x),$$

т.е. $\left\langle a(x)\cdot b(x)\right|_{p(x)} = c(x)$.

Представим перемножаемые элементы конечного поля в виде многочленов

$$a = a_{m-1}\alpha^{m-1} + a_{m-2}\alpha^{m-2} + \ldots + a_1\alpha^1 + a_0\alpha^0,$$

$$b = b_{m-1}\alpha^{m-1} + b_{m-2}\alpha^{m-2} + \ldots + b_1\alpha^1 + b_0\alpha^0,$$

где α – примитивный элемент поля, корень неприводимого многочлена $p(x)$.

Результатом перемножения этих многочленов будет многочлен

$$c = e_{2m-2}\alpha^{2m-2} + e_{2m-3}\alpha^{2m-3} + \ldots + e_{m-1}\alpha^{m-1} + e_m\alpha^m + e_{m-1}\alpha^{m-1} + e_{m-2}\alpha^{m-2} + \ldots + e_1\alpha^1 + e_0\alpha^0, \quad (1)$$

где $e_l = \sum_{i+j=l} a_i b_j$ $(l = \overline{0, 2m-2})$.

Проведя преобразование выражения (1) с учетом того, что $\alpha^m = h(\alpha) = p_{m-1}\alpha^{m-1} + \ldots + p_0 a^0$, получим

$$c = \sum_{i=0}^{m-1}(e_i + \sum_{j=0}^{m-2} p_i^{(j)} e_{m+j})\alpha^i, \quad (2)$$

где $e_i = \sum_{u=0}^{i} a_{i-u} b_u$, $e_{m+j} = \sum_{u=1}^{m-1-j} a_{m-u} b_{j+u}$,

$p_i^{(j)} = \sum_{i=1}^{j} p_{m-1} p_i^{(j-1)} + p_{i-j}$ (при $j = 0: p_i^{(0)} = p_i$;

при $i < j: p_{i-j} = 0$).

Назовем коэффициенты $p_i^{(j)}$ *операционными коэффициентами* конечного поля.

Метод умножения элементов конечного поля $GF(2^m)$ в соответствии с выражением (2), основанный на умножении многочленов над полем $GF(2)$ с необходимостью вычисления операционных коэффициентов $p_i^{(j)}$, назовем *методом непосредственного умножения элементов конечного поля*.

Представим выражение (2) в матричной форме

$$\begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{m-2} \\ c_{m-1} \end{bmatrix} = \begin{bmatrix} e_0 \\ e_1 \\ \vdots \\ e_{m-2} \\ e_{m-1} \end{bmatrix} \oplus \begin{bmatrix} p_0^{(0)} & p_0^{(1)} & \ldots & p_0^{(m-2)} \\ p_1^{(0)} & p_1^{(1)} & \ldots & p_1^{(m-2)} \\ \vdots & \vdots & \ldots & \vdots \\ p_{m-2}^{(0)} & p_{m-2}^{(1)} & \ldots & p_{m-2}^{(m-2)} \\ p_{m-1}^{(0)} & p_{m-1}^{(1)} & \ldots & p_{m-1}^{(m-2)} \end{bmatrix} \otimes \begin{bmatrix} e_m \\ e_{m+1} \\ \vdots \\ e_{2m-3} \\ e_{2m-2} \end{bmatrix} =$$

$$= E_1 \oplus P \otimes E_2, \quad (3)$$

где величины e_l $(l = \overline{0, 2m-2})$ и $p_i^{(j)}$ определяются соответственно из выражений

$$\begin{bmatrix} e_0 \\ e_1 \\ \vdots \\ e_{m-2} \\ e_{m-1} \end{bmatrix} = \begin{bmatrix} a_0 & & & & \\ a_1 & a_0 & & \ddots & \\ \vdots & \vdots & & & \\ a_{m-2} & a_{m-3} & \ldots & a_0 & \\ a_{m-1} & a_{m-2} & \ldots & a_1 & a_0 \end{bmatrix} \otimes \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{m-2} \\ b_{m-1} \end{bmatrix} = A_1 \otimes \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{m-2} \\ b_{m-1} \end{bmatrix},$$

$$\begin{bmatrix} e_m \\ e_{m+1} \\ \vdots \\ e_{2m-3} \\ e_{2m-2} \end{bmatrix} = \begin{bmatrix} a_{m-1} & a_{m-2} & \dots & a_2 & a_1 \\ & a_{m-1} & \ddots & a_3 & a_2 \\ & & & \vdots & \vdots \\ & & & a_{m-1} & a_{m-2} \\ & & & & a_{m-1} \end{bmatrix} \otimes \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{m-2} \\ b_{m-1} \end{bmatrix} = A_2 \otimes \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{m-2} \\ b_{m-1} \end{bmatrix} \quad (4)$$

и

$$p_i^{(j)} = \begin{bmatrix} p_i^{(0)} \\ p_i^{(1)} \\ p_i^{(2)} \\ \vdots \\ p_i^{(m-3)} \\ p_i^{(m-2)} \end{bmatrix} =$$

$$= \begin{bmatrix} p_i \\ p_{i-1} \oplus p_{m-1} \otimes p_i^{(0)} \\ p_{i-2} \oplus p_{m-1} \otimes p_i^{(1)} \oplus p_{m-2} \otimes p_i^{(0)} \\ \vdots \\ p_{i-(m-3)} \oplus p_{m-1} \otimes p_i^{(m-4)} \oplus p_{m-2} \otimes p_i^{(m-5)} \oplus \dots \oplus p_3 \otimes p_i^{(0)} \\ p_{i-(m-2)} \oplus p_{m-1} \otimes p_i^{(m-3)} \oplus p_{m-2} \otimes p_i^{(m-4)} \oplus \dots \oplus p_3 \otimes p_i^{(1)} \oplus p_2 \otimes p_i^{(0)} \end{bmatrix}. \quad (5)$$

Здесь $\otimes$, $\oplus$ – модульные операции умножения и сложения соответственно.

Матрицу (P) операционных коэффициентов ($p_i^{(j)}$) назовем *операционной матрицей* конечного поля.

Если принять, что w – вес многочлена $h(\alpha)$, то операционная матрица P будет иметь не менее $(m-k)(m-w)$ нулевых коэффициентов $p_i^{(j)}$. Это необходимо учитывать при определении сложности комбинационных схем умножения элементов конечного поля (КСУЭ) для случая, когда коэффициенты $p_i^{(j)}$ вычисляются заранее и их величины заложены в структуру КСУЭ.

Алгоритм непосредственного умножения элементов конечного поля

Умножение элементов поля $GF(2^m)$ в соответствии с выражением (3) можно выполнять в три этапа по следующему алгоритму, который назовем *алгоритмом непосредственного умножения элементов конечного поля* (*алгоритм У*2):

• Выполняется умножение многочленов над полем $GF(2)$, представляющих элементы поля $GF(2^m)$. Результатом умножения будет многочлен, коэффициенты которого можно записать в виде вектора $(e_{2m-2}, \dots, e_m, e_{m-1}, \dots, e_0)$. Умножение можно выполнять в устройстве умножения, рассчитанном на получение результата с удвоенным количеством разрядов.

• Выполняется умножение операционной матрицы P на вектор старших разрядов (e_m, $e_{m-1}, \dots, e_{2m-3}, e_{2m-2}$) полученного результата. Получим вектор $(\tilde{e}_{m-1}, \tilde{e}_{m-2}, \dots, \tilde{e}_1, \tilde{e}_0)$.

• Проводится поразрядное сложение вектора $(e_{m-1}, e_{m-2}, \dots, e_1, e_0)$, полученного на первом этапе вычисления, и вектора $(\tilde{e}_{m-1}, \tilde{e}_{m-2}, \dots, \tilde{e}_1, \tilde{e}_0)$, полученного на втором этапе. В итоге имеем результат умножения двух элементов конечного поля – $(c_{m-1}, c_{m-2}, \dots, c_1, c_0)$.

Выражения (3)–(5) могут быть реализованы с помощью комбинационных схем. Покажем, как это можно сделать.

Схемы вычисления операционных коэффициентов $p_i^{(j)}$

Схема устройства вычисления операционных коэффициентов $p_i^{(j)}$ (СВОК) показана на рис. 1 и содержит $(m-1)(m-2)/2$ схем И и столько же сумматоров по модулю 2. Для одновременного вычисления всех $(m-1)$ i-х коэффициентов $p_i^{(j)}$ необходимо иметь m таких схем. Значит, аппаратурная сложность схемы устройства вычисления операционных коэффициентов следующая

$N_{СВОК}^{B} \le 5m(m-1)(m-2)/2$ – верхняя граница.

Схемы каждого из разрядов имеют разное количество сумматоров по модулю 2. Так в схеме первого разряда на $(m-2)$ сумматоров меньше, чем в схеме, представленной на рис. 1; в схеме второго разряда – меньше на $(m-3)$; в схеме третьего разряда – на $(m-4)$; …; в схеме $(m-1)$-го разряда – меньше на единицу; в схеме m-го разряда – одинаковое количество сумматоров, т.е. по совокупности потребуется меньше сумматоров по модулю 2 на величину $(m-1)^2/2$. Учитывая это, имеем

$N_{\text{СВОК}}^{\text{Н}} \ge (m-1)[m(5m-14)+4]/2$ – нижняя граница.

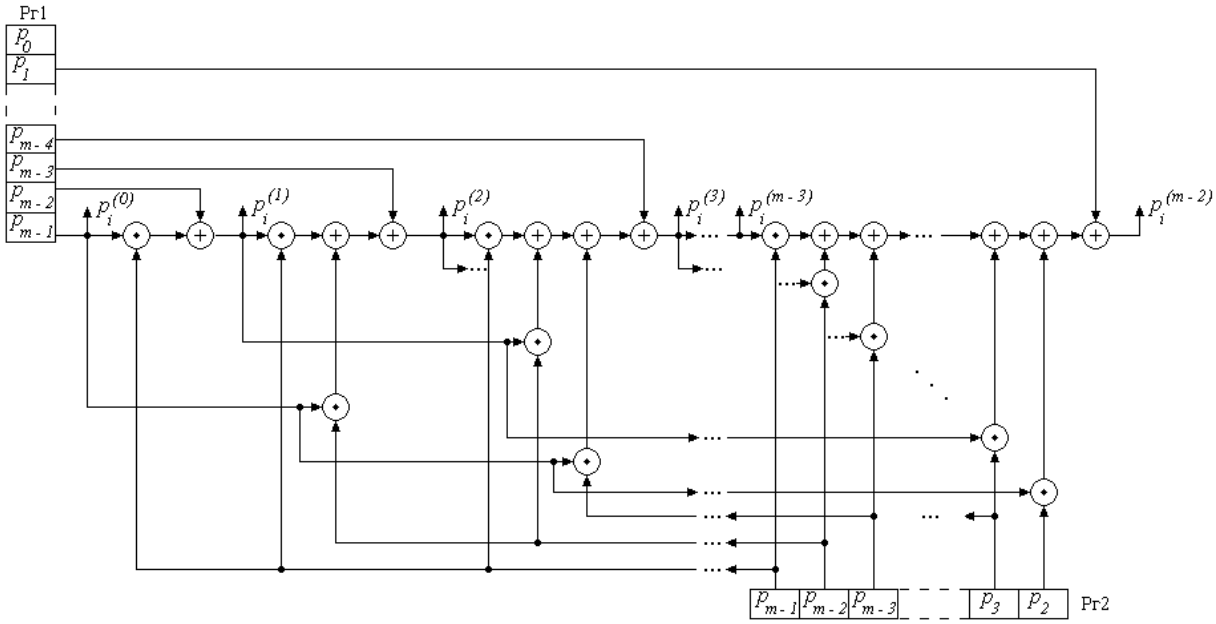

Рис. 1. Схема одного (m – 1)-го разряда устройства вычисления операционных коэффициентов $p_i^{(j)}$

Все значения коэффициентов $p_i^{(j)}$ будут получены через время

$$T_{\text{СВОК}} = (m-2)(3m-1)t/2.$$

Регистры Рг1 и Рг2 не есть частью СВОК и при расчете сложности схемы не учитываются. Если потребуется запоминать все вычисленные значения $p_i^{(j)}$, то необходимо будет иметь $m(m-1)$ ячеек памяти.

Устройство, построенное на основе приведенной схемы, позволяет вычислять коэффициенты $p_i^{(j)}$ для общего случая с возможностью изменения неприводимого многочлена $p(x)$ заданной степени m. Это устройство путем наращивания легко адаптируемо к увеличению степени m неприводимого многочлена. Для этого необходимо дополнительно сделать схему для m-го разряда, вычисляющую величины $p_m^{(j)}$, а также к выходам $p_i^{(m-2)}$ каждой из имеющихся схем добавить соответствующие блоки, на выходах которых будем получать значения $p_i^{(m-1)}$. При уменьшении степени m никаких изменений можно не проводить.

Комбинационные схемы умножения

Комбинационная схема умножения (КСУЭ-У2), реализующая алгоритм умножения в соответствии с выражением (3) (алгоритм У2), имеет три уровня.

На первом уровне КСУЭ-У2 вычисляются величины e_l ($l = \overline{0, 2m-2}$). Вычисление проводится с использованием комбинационных схем умножения многочленов над полем $GF(2)$ (КСУМ) [9].

На втором уровне КСУЭ-У2 реализуется произведение матриц $P \otimes E_2$, т.е. вычисляются величины $\tilde{e}_i$ ($i = \overline{0, m-1}$). Этот уровень составляет группа блоков $G_2 = \{B_0, B_1, \ldots, B_{m-1}\}$.

Для варианта построения схемы с возможностью изменения многочлена $p(x)$ и хранением величин $p_i^{(j)}$ эта группа блоков содержит $m(m-1)$ схем И и $m(m-2)$ сумматоров по модулю 2. Здесь схемы И необходимы для того, чтобы иметь возможность изменять многочлен $p(x)$.

При фиксированном многочлене $p(x)$ схемы И не нужны, так как коэффициенты $p_i^{(j)}$ заложены в структуру схемы умножения матриц $P \otimes E_2$.

На третьем уровне КСУЭ-У2 реализуется матрица C как сумма векторов e_i и $\tilde{e}_i$ ($i = \overline{0, m-1}$). Этот уровень составляет группа блоков $G_3 = \{C_0, C_1, \ldots, C_{m-1}\}$, которая содержит m сумматоров по модулю 2.

КСУЭ-У2 для варианта построения с возможностью изменения многочлена $p(x)$ и хранением величин $p_i^{(j)}$ приведена на рис. 2. Здесь не показана схема вычисления коэффициентов $p_i^{(j)}$.

Временная сложность КСУЭ-У2 составляет (без учета сложности вычисления операционных коэффициентов $p_i^{(j)}$ и их хранения):

$$T^{\text{И}}_{\text{КСУЭ-У2}} = (3m+2)t\,.$$

При построении КСУЭ для любого неприводимого полинома $p(x)$ заданной степени можно добиться однородности и универсальности структуры схемы. Для этого в качестве базовой ячейки (БЯ) выбирается функциональная ячейка (ФЯ), состоящая из одного двухвходового элемента И и сумматора по модулю 2.

Построенная таким образом схема будет называться универсальной КСУЭ (УКСУЭ).

Универсальная КСУЭ (УКСУЭ-У2), реализующая выражение (3) и показанная на рис. 3, как и КСУЭ-У2, имеет три уровня.

Первый уровень УКСУЭ-У2 представляет собой УКСУМ, состоящую из групп блоков $G_1^{(1)} = \{E_0, E_1, \ldots, E_{m-1}\}$ и $G_1^{(2)} = \{E_m, E_{m+1}, \ldots, E_{2m-2}\}$. Каждый из блоков E_l ($l = \overline{0, 2m-2}$) имеет одну (E_0 и E_{2m-2}) или нескольких БЯ.

Второй уровень УКСУЭ-У2 составляют блоки B_i ($i = \overline{0, m-1}$), каждый из которых содержит $(m-1)$ БЯ. Для варианта построения схемы с возможностью изменения многочлена $p(x)$ и хранением величин $p_i^{(j)}$ этот уровень содержит $m(m-1)$ схем И и $m(m-1)$ сумматоров по модулю 2.

Третий уровень УКСУЭ-У2 составляют блоки C_i ($i = \overline{0, m-1}$), каждый из которых имеет по одной БЯ.

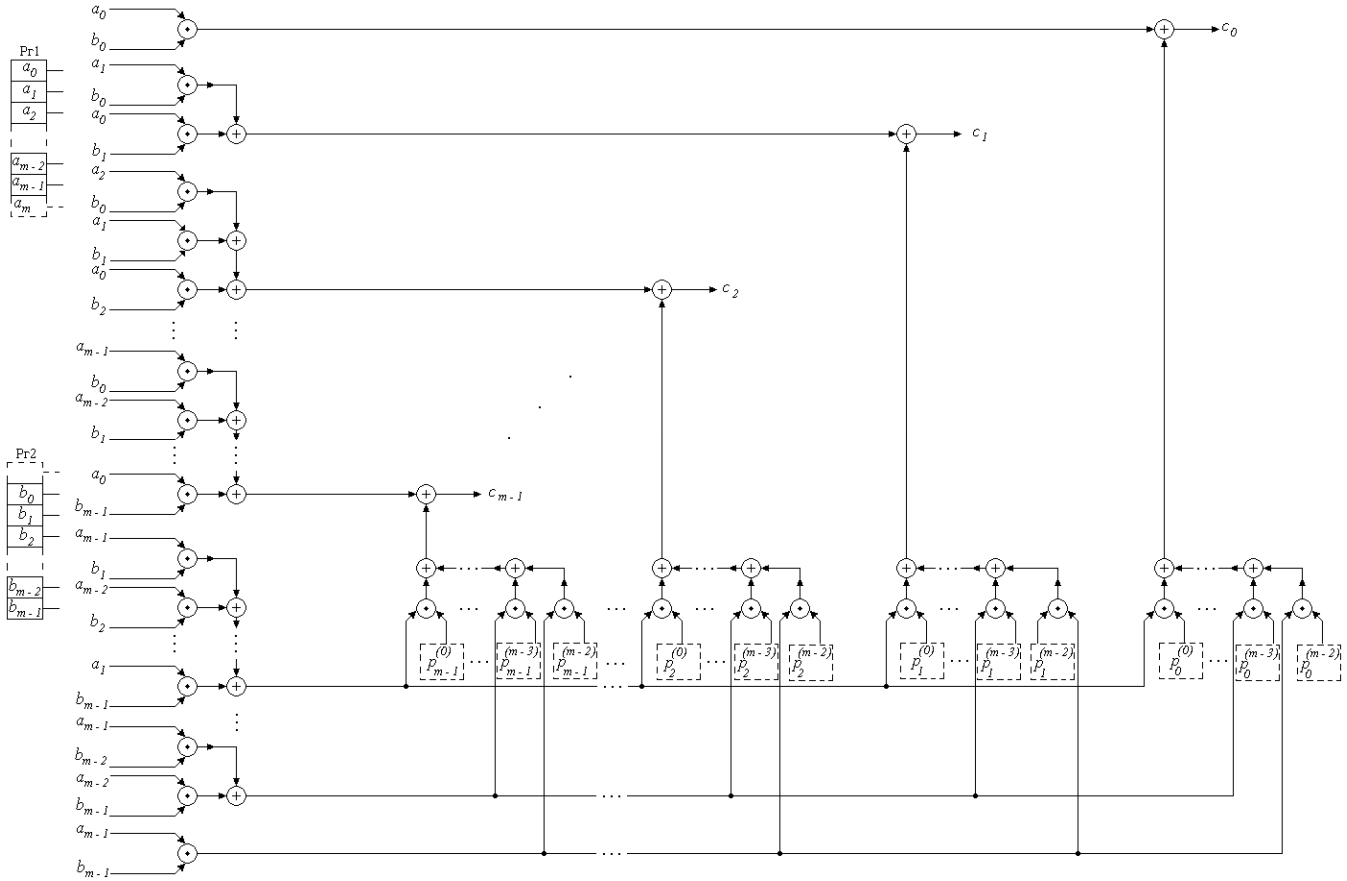

Рис. 2. Комбинационная схема умножения элементов конечного поля $GF(2^m)$ (КСУЭ-У2)

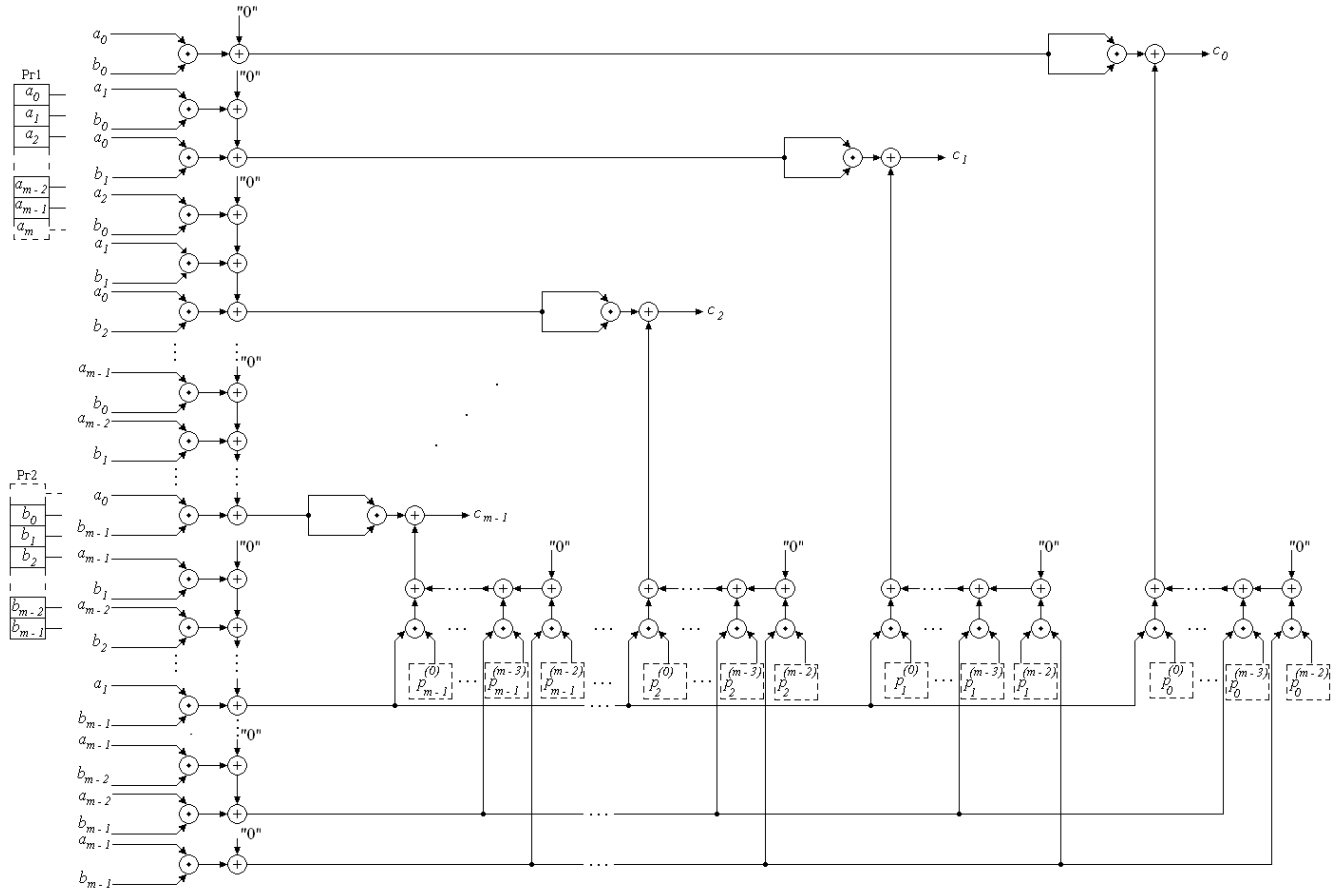

Рис. 3. Универсальная комбинационная схема умножения элементов конечного поля $GF(2^m)$ (УКСУЭ-У2)

Структура УКСУЭ-У2 при наличии БЯ позволяет не только менять вид многочлена $p(x)$ заданной степени с предварительным вычислением величин $p_i^{(j)}$, но и строить УКСУЭ-У2 при изменении степени многочлена $p(x)$. Если степень многочлена $p(x)$ меньше m, то схему можно не менять. Покажем, как наращивается структура УКСУЭ-У2 при увеличении степени многочлена $p(x)$.

Пусть $\deg p(x) = m+1$. Тогда

$$C' = \begin{bmatrix} c'_m \\ c'_{m-1} \\ \vdots \\ c'_1 \\ c'_0 \end{bmatrix} = \begin{bmatrix} e'_m \\ e'_{m-1} \\ \vdots \\ e'_1 \\ e'_0 \end{bmatrix} \oplus .$$

$$\oplus \begin{bmatrix} p_m^{(0)} & p_m^{(1)} & \dots & p_m^{(m-2)} & p_m^{(m-1)} \\ p_{m-1}^{(0)} & p_{m-1}^{(1)} & \dots & p_{m-1}^{(m-2)} & p_{m-1}^{(m-1)} \\ \vdots & \vdots & \dots & \vdots & \vdots \\ p_1^{(0)} & p_1^{(1)} & \dots & p_1^{(m-2)} & p_1^{(m-1)} \\ p_0^{(0)} & p_0^{(1)} & \dots & p_0^{(m-2)} & p_0^{(m-1)} \end{bmatrix} \otimes \begin{bmatrix} e'_{m+1} \\ e'_{m+2} \\ \vdots \\ e'_{2m-1} \\ e'_{2m} \end{bmatrix}.$$

Операционные коэффициенты $p_i^{(j)}$ принимают новые значения.

Изменения в Рг1 и Рг2, а также в группах $G_1^{(1)}$ и $G_1^{(2)}$ первого уровня УКСУЭ-У2 проводится так же, как в УКСУМ [9].

В группе G_2 второго уровня УКСУЭ-У2 добавляется один новый блок E'_m, состоящий из m БЯ; в каждом из блоков $B_0, B_1, \dots, B_{m-1}$ становится на одну БЯ больше.

Группа G_3 третьего уровня УКСУЭ-У2 дополняется одним новым блоком C'_m, состоящим из одной БЯ.

Значит, для получения новой УКСУЭ-У2 при $\deg p(x) = m+1$ необходимо:

- регистр Рг1 дополнить одной ячейкой со стороны старших разрядов, а регистр Рг2 – одной ячейкой со стороны младших разрядов;
- в каждом из блоков E_l ($l = \overline{0, 2m-2}$) групп $G_1^{(1)}$ и $G_1^{(2)}$ к свободному входу сумматора по

модулю 2 подключить по одной БЯ. Группы $G_1^{(1)}$ и $G_1^{(2)}$ дополнить соответственно $E'_0 =$ БЯ и $E'_{2m} =$ БЯ. Получим новые группы $G_1^{(1)} = \{E'_0, E'_1, \dots, E'_m\}$ и $G_1^{(2)} = \{E'_m, E'_{m+1}, \dots, E'_{2m}\}$;

• в каждом из блоков B_i $(i = \overline{0, m-1})$ группы G_2 к свободному входу сумматора по модулю 2 подключить по одной БЯ. Группу G_2 дополнить блоком $B'_m = m \cdot$БЯ. Получим новую группу $G_2 = \{B'_0, B'_1, \dots, B'_m\}$;

• группу G_3 дополнить блоком $C'_m =$ БЯ;

• каждый из регистров Рг3÷Рг(m+1), предназначенных для хранения величин $p_i^{(j)}$, дополнить одним разрядом, выход которого подключается к входу дополнительной БЯ блоков группы G_2;

• подключить новые БЯ групп $G_1^{(1)}$ и $G_1^{(2)}$ к соответствующим выходам регистров Рг1 и Рг2;

• подключить блок E'_{2m} группы $G_1^{(2)}$ к каждому из блоков B'_i группы G_2;

• подключить регистр Рг(m+2) и блоки группы $G_1^{(2)}$ к блоку B'_m группы G_2;

• подключить блок E'_0 группы $G_1^{(1)}$ и блок B'_m группы G_2 к блоку C'_m группы G_3.

Сравнение схем умножения элементов конечного поля

Сравнение КСУЭ-У2 с ССУЭ показывает, что временная сложность КСУЭ-У2 всегда меньше временной сложности ССУЭ на величину:

$$\Delta T = T^{\text{И}}_{\text{ССУЭ}} - T^{\text{И}}_{\text{КСУЭ-У2}} = (19m + 13)t\,.$$

Сравнивая КСУЭ-У2 с БШУЭ, получаем минимальное (т.к. схема вычисления величин $\alpha_{kl}^{(i)}$ и ее сложность в публикациях не приводятся) значение величины временной сложности:

$$\Delta T^{\min} = T^{\text{И}}_{\text{БШУЭ}} - T^{\text{И}}_{\text{КСУЭ-У2}} = 3(m-2)t\,,$$

т.е. временная сложность КСУЭ-У2 меньше временной сложности БШУЭ для всех практически применяемых конечных полей.

Заключение. Предложенный метод умножения элементов конечного поля $GF(2^m)$, основанный на умножении многочленов над полем $GF(2)$, позволяет создавать схемы устройств, структура которых однородна, универсальна и легко наращиваема при увеличении степени образующего поле неприводимого многочлена $p(x)$. Такая структура делает эти устройства перспективными для реализации в виде БИС и выгодно отличает их от линейных последовательностных машин. Схемы устройств, реализующих данный метод, более быстродействующие в сравнении с известными схемами.

1. *Питерсон У.*, *Уэлдон Э.* Коды, исправляющие ошибки / Под ред. Р.Л. Добрушина и С.И. Самойленко. – М.: Мир, 1976. – 596 с.
2. *Гилл А.* Линейные последовательностные машины / Под ред. Я.З. Цыпкина. – М.: Наука, 1974. – 287 с.
3. *Жуков И.А.*, *Кубицкий В.И.* Оценка сложности вычислений в конечных полях // Інформаційні технології та комп'ютерна інженерія: Міжнародний науково-технічний журнал. – 2013. – Т. 2, № 27. – С. 21–27.
4. *Кубицкий В.И.* Методы вычислений в конечных полях // Проблеми інформатизації та управління: Зб. наук. праць. – 2009. – **4** (28). – С. 88–98.
5. *Bartee T.C.*, *Shneider D.I.* Computations with finite fields // Information and Control. – 1963. – **6**. – P. 79–98.
6. *Кларк Дж.*, *мл.*, *Кейн Дж.* Кодирование с исправлением ошибок в системах цифровой связи / Под ред. Б.С. Цыбакова. – М.: Радио и связь, 1987. – 391 с.
7. *Сюрин В.Н.*, *Иванов Н.Н.*, *Альхимович В.В.* Реализация вычислений в конечных полях // Зарубежная электроника. – 1990. – № 5. – С. 59–68.
8. *Мак-Вильямс Ф. Дж.*, *Слоэн Н. Дж. А.* Теория кодов, исправляющих ошибки / Под ред. Л.А. Бассалыго. – М.: Связь, 1979. – 744 с.
9. *Кубицкий В.И.* Операции над многочленами в поле $GF(2)$ // Науч. вестн. ГосНИИ «Аэронавигация». – 2007. – № 7. – С. 185–194.

Поступила 11.12.2013

Тел. для справок: +38 044 497-7257, +7 962 912-2920 (Киев, Москва)
E-mail: *zhuia@ukr.net*, *vkubitski@mail.ru*