УДК 681.325.3:621.391.25:621.394.14

Кубицкий В.И.

МЕТОДЫ ВЫЧИСЛЕНИЙ В КОНЕЧНЫХ ПОЛЯХ

ГосНИИ «Аэронавигация» (Россия, Москва)

Проведен анализ методов вычислений в конечных полях. Рассматрены операции над многочленами в поле GF(2) и над элементами конечных полей $GF(2^m)$. Определены аппаратурная и временная сложности схем, реализующих эти операции на основе известных методов вычислений в конечных полях. Дана их сравнительная оценка

Вычисления в конечных полях (или полях Галуа) используются во многих областях науки и техники. Операции в конечных полях лежат в основе алгоритмов кодирования и декодирования циклических кодов. На основе этих операций также могут быть реализованы некоторые алгоритмы цифровой обработки сигналов, такие как вычисление цифровой свертки или дискретного преобразования Фурье. Вычисления в конечных полях используются в целях повышения точности действия радиолокационных станций, для модуляции выходных сигналов которых применяются *M*-последовательности. Теория конечных полей также может использоваться в криптографических преобразованиях с открытыми ключами (типа *RSA*) на сети общего пользования со многими абонентами.

Вопросы реализации вычислений в конечных полях рассматриваются разрозненно, публикации по этим вопросам немногочисленны. Все это препятствует внедрению современных методов кодирования/декодирования и преобразования сигналов в инженерную практику. Однако, в виду важности конечных полей для реализации современных радиоэлектронных систем самого разного назначения, необходимо совершенствовать технику конечных полей.

Учитывая возможности современной микроэлектроники, следует отметить, что перспективные алгоритмы выполнения операций в конечных полях должны быть не только эффективными, но и технологичными, обязательно реализуемыми на базе БИС и должны обеспечивать однородность схем реализации.

Набор команд современных компьютеров не приспособлен для выполнения операций над элементами конечных полей. Поэтому для таких операций создаются дополнительные подпрограммы или специальные вычислительные устройства.

Вычисления в конечных полях с помощью вычислительных устройств могут быть реализованы различными способами. Наибольшее распространение получила реализация операций над многочленами и элементами конечных полей в регистрах сдвига [1 - 3].

Для выполнения операций в конечных полях можно использовать процедуры, предложенные в [4]. В [5] предлагается применять матричные вычислительные устройства. Вычисления можно производить также с использованием табличных методов.

Некоторые алгоритмы вычислений в полях Галуа могут быть реализованы только программным способом. Так, если для криптографических преобразований используются поля высоких порядков $GF(2^{127})$ или $GF(2^{400})$, то логарифм, представляющий элемент поля, вычисляется с помощью специальных алгоритмов [6], которые реализуются программными методами с применением высокопроизводительных компьютеров. Эти алгоритмы либо требуют значительного времени вычислений, либо приводят к возрастанию вероятности ложного определения логарифма, что практически может привести к искажению сообщений.

Для большинства информационно-вычислительных систем и систем связи значительную роль играют такие параметры как аппаратурная сложность со-

ставляющих их элементов и время обработки/передачи информации. С точки зрения улучшения этих параметров необходимо рассматривать реализацию операций в конечных полях.

Рассмотрим некоторые известные способы и схемы выполнения операций над многочленами в поле $GF(2)$ и над элементами конечных полей $GF(2^m)$ и приведём аппаратурные и временные сложности этих схем.

Под *аппаратурной сложностью* (N) схемы будем понимать количество функциональных элементов базисного набора (*И*, *ИЛИ*, *НЕ*) в схеме, реализующей заданную функцию.

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

1. Умножение многочленов над полем GF(2)

Операцию умножения будем проводить с многочленами

$$\begin{aligned} a(x) &= a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \ldots + a_1 x + a_0 \\ b(x) &= b_{n-1}x^{n-1} + b_{n-2}x^{n-2} + \ldots + b_1 x + b_0 \end{aligned}, \quad (1)$$

где a_i, $b_j \in GF(p)$.

При умножении в качестве множимого используется многочлен $a(x)$, в качестве множителя – многочлен $b(x)$. Результатом умножения будет многочлен:

$$e(x) = \sum_{i=0}^{(m-1)+(n-1)} e_i x^i ,$$

где $e_i = \sum_{j=0}^{i} a_j b_{i-j}$.

1.1. Умножение многочленов на сдвиговых регистрах.

Схемы, реализующие умножение многочленов $a(x)$ и $b(x)$ на сдвиговых регистрах, приводятся в [1 - 3, 7, 8 и др.]. Для таких схем введём аббревиатуру ССУМ и определим их сложности.

Для умножения двух многочленов необходимо иметь: сдвиговой регистр, регистр для записи и хранения одного из многочленов-сомножителей (может отсутствовать), набор логических элементов *И* и набор сумматоров по модулю 2. Кроме того, требуется схема управления сдвигами, которая должна определять количество сдвигов (пропорциональное сумме степеней многочленов и равное $(\deg a(x)+1)+(\deg b(x)-1)=m+(n-2)$) и выдавать команды на выполнение сдвигов. При определении сложности устройств умножения многочленов схема управления сдвигами рассматриваться не будет.

Коэффициенты одного из многочленов-сомножителей (пусть это будет многочлен $b(x)$) записываются в регистр для записи и хранения и подаются на умножение параллельно. Коэффициенты другого многочлена-сомножителя (многочлен $a(x)$) поступают в регистр сдвига последовательно.

Из [9] известно, что простейшей схемой регистра сдвига является схема, построенная на D-триггерах. Отметим, что перед операцией умножения требуется выполнять обнуление регистра сдвига, для чего необходимо иметь соответствующие входы в триггерах. Регистр для записи и хранения многочлена $b(x)$ может быть построен на однотактных триггерах и предварительного обнуления не требует. Поэтому сложность реализации умножения многочленов будем определять для схемы, построенной с использованием таких триггеров.

С учетом этого в результате расчетов получим, что аппаратурная (N) и временная (T) сложность ССУМ равна:

$$N_{ССУМ} = 34n - 23 ,$$

$$T_{ССУМ} = [20(m+n) - 29]t .$$

Здесь не учитывается время обнуления регистра сдвига, которое равно $4t$, так как обнуление можно производить во время записи коэффициентов b_i в соответствующий регистр (при его наличии) и

время этой записи больше времени обнуления (равно $7t$).

Та как в ССУМ может отсутствовать регистр для записи и хранения одного из многочленов-сомножителей (многочлена $b(x)$), то при определении времени умножения необходимо учитывать время обнуления регистра сдвига.

Тогда сложность ССУМ будет следующей:

$$N_{ССУМ} = 25n - 23,$$
$$T_{ССУМ} = [20(m+n) - 32]t.$$

1.2. Умножение многочленов на матричных вычислительных устройствах.

В [5] предлагается производить умножение многочленов над полем $GF(2)$ с помощью так называемых матричных вычислительных устройств (МВУ). Предложенный подход позволяет повысить быстродействие по сравнению с умножением на сдвиговых регистрах. Матричные устройства умножения многочленов (МУУМ) на базе МВУ строятся на основе базовой ячейки (БЯ), состоящей из одного двухвходового элемента *И* и сумматора по модулю 2.

Сложность МУУМ составляет [5]:

$$N_{МУУМ} = 5mn, \quad T_{МУУМ} = (3n+1)t.$$

2. Деление многочленов над полем GF(2)

Деление многочлена $c(x)$ на многочлен $a(x)$ производится в соответствии с выражением:

$$c(x) = a(x) \cdot b(x) + r(x),$$

где многочлены $a(x)$ и $b(x)$ представлены выражениями (1), многочлен $r(x) = r_{r-1}x^{r-1} + r_{r-2}x^{r-2} + \ldots + r_1 x + r_0$ является остатком от деления, коэффициенты a_i, b_j, $r_s \in GF(p)$.

Степень многочлена $c(x)$ (делимого) равна:

$$\deg c(x) = (d-1) =$$
$$= \deg a(x) + \deg b(x) = (m-1) + (n-1).$$

Степень многочлена $b(x)$ (частного) изначально неизвестна, но ее можно вычислить из выражения:

$$\deg b(x) = (n-1) =$$
$$= \deg c(x) + \deg a(x) = (d-1) + (m-1).$$

Степень многочлена $r(x)$ (остатка) определяется степенью делителя $a(x)$:

$$\deg r(x) = (r-1) < \deg a(x) < (m-1).$$

2.1. Деление многочленов на сдвиговых регистрах.

Схемы, реализующие деление многочленов $c(x)$ и $a(x)$ на сдвиговых регистрах, приводятся в [1 - 3, 7 и др.]. Для таких схем введём аббревиатуру ССДМ и определим их сложности.

Для деления, также как и для умножения двух многочленов, необходимо иметь: сдвиговой регистр, регистр для записи и хранения делителя (может отсутствовать), набор логических элементов *И* и набор сумматоров по модулю 2. Кроме того, требуется еще схема управления сдвигами, которая должна определять количество сдвигов (пропорциональное сумме степеней многочленов и равное $(\deg c(x)+1)+(\deg a(x)-1) = d+(m-2)$) и выдавать команды на выполнение сдвигов. При определении сложности устройств деления многочленов схема управления сдвигами рассматриваться не будет.

Коэффициенты делителя $a(x)$ записываются в регистр для записи и хранения делителя и подаются для деления параллельно. Коэффициенты делителя $c(x)$ поступают в регистр сдвига последовательно.

Регистр сдвига строится на *D*-триггерах, регистр для записи и хранения делителя $a(x)$ – на однотактных триггерах.

Сложность устройств деления многочленов (ССДМ), построенных на сдвиговых регистрах составляет:

$$N_{ССДМ} = 34m - 33,$$
$$T_{ССДМ} = (20d+7)t.$$

Здесь не учитывается время обнуления регистра сдвига, которое равно $4t$, так как обнуление можно производить во время записи коэффициентов a_i в соответствующий регистр (при его наличии) и время этой записи больше времени обнуления (равно $7t$).

Так как в ССДМ может отсутствовать регистр для записи и хранения делителя $a(x)$, то сложность ССДМ будет следующей:

$$N_{ССДМ} = 25m - 24,$$
$$T_{ССДМ} = (20d + 4)t.$$

Здесь при определении времени деления учитывалось время обнуления регистра сдвига.

2.2. Деление многочленов на матричных вычислительных устройствах.

Сложность устройств деление многочленов (МУДМ) над полем $GF(2)$, построенных на базе МВУ, определяется разрядностями делимого и делителя и составляет [5]:

$$N_{МУДМ} = 5(d - m + 1)m,$$
$$T_{МУДМ} = 4(d - m + 1)t.$$

3. Операции над элементами конечного поля GF(2^m)

Пусть:

$GF(p^m)$ – конечное поле, порождаемое неприводимым примитивным многочленом

$$p(x) = x^m + p_{m-1}x^{m-1} + \ldots + p_0x^0 = x^m + h(x)$$

степени m;

α – примитивный элемент поля, корень неприводимого примитивного полинома $p(x)$.

Здесь $p_i \in GF(p)$, где p – простое число, характеристика поля.

Тогда любой элемент конечного поля $GF(p^m)$ может быть представлен в виде:

– степени примитивного элемента α^i ($i = \overline{0,\ \ 2^m - 2}$);

– логарифма (индекса) i элемента α^i ($\alpha^i \in GF(p^m)$);

– многочлена $h(\alpha) = p_{m-1}\alpha^{m-1} + \ldots + p_0\alpha^0$;

– m–мерного p-ичного вектора $[p_{m-1}, p_{m-2}, \ldots, p_0]$.

Соответственно представления элементов конечного поля называются: степенное, логарифмическое, полиномиальное, векторное.

В зависимости от способа представления элементов конечного поля применяются различные способы реализации вычислений в конечном поле. Для случая представления элементов поля в виде степеней примитивного элемента α конечного поля используются логарифмы Зеча. Если элементы конечного поля представлены в виде логарифма (индекса) i элемента α^i, то используются таблицы логарифмов и антилогарифмов. Полиномиальное и векторное представление элементов конечного поля позволяет выполнять операции над этими элементами с применением следующих способов:

– непосредственное вычисление с использованием логических функций;

– вычисление на сдвиговых регистрах;

– вычисление в матричных вычислительных устройствах;

– реализация умножителя в ПЗУ.

Для полиномиального и векторного представления элементов конечного поля сложение элементов выполняется проще, чем для других способов представления.

Кратко охарактеризуем упомянутые способы реализации вычислений в конечных полях $GF(2^m)$.

1) *Непосредственное вычисление с использованием логических функций.*

В этом случае вычисление может выполняться с использованием двухвходовых логических элементов *И* и сумматоров по модулю 2. Характерным примером для этого способа является способ, предложенный в [4] и позволяющий реализовывать операции умножения, а также инвертирования в конечных полях малых порядков (со степенью порождающего полинома не более 6), на комбинацион-

ных схемах. Для инвертирования элементов конечных полей более высоких порядков необходимо использовать компьютеры. Этот способ обеспечивает высокую скорость вычислений.

2) *Вычисление на сдвиговых регистрах.*

Схемы, реализующие этот способ вычисления, находят широкое применение и описаны во многих литературных источниках ([1 - 3, 7, 10, 11 и др.]). Однако, скорость вычисления для данного способа меньше, чем для способа непосредственного вычисления с использованием логических функций. Но, как показывают исследования, при больших m в некоторых случаях (скажем, при $m \geq 9$ для изменяемого неприводимого полинома $p(x)$) этот способ может привести к экономии аппаратных средств. При такой реализации для большинства практически применимых конечных полей время выполнения операций над элементами этих полей не обеспечивает возможности роста скоростей передачи информации.

3) *Реализация умножителя в ПЗУ.*

Для полей $GF(2^m)$ с небольшим числом элементов (скажем, для $m \leq 6$) более эффективным по сравнению со способом непосредственного вычисления произведения с использованием логических функций является умножение в ПЗУ. В этом случае результаты произведений элементов поля и результаты их мультипликативных инверсий должны содержаться в таблицах, для хранения которых можно использовать постоянные запоминающие устройства (ПЗУ) или программируемые логические матрицы (ПЛМ). Очевидно, что в этом случае необходимо заранее вычислять результаты произведений элементов поля и результаты их мультипликативных инверсий перед тем как занести их в таблицу. Кроме этого, такая таблица будет отражать результаты операций 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 2^{2m} x m.

4) *Использование таблиц логарифмов и антилогарифмов.*

При этом способе 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.

В отличие от логарифма действительного числа логарифм в конечном поле является чрезвычайно нерегулярной функцией. Неизвестно никакого быстрого способа нахождения логарифма примитивного элемента α конечного поля: либо его находят непосредственно, вычисляя подряд все последовательные степени примитивного элемента α до тех пор, пока не встретится нужный элемент конечного поля (что очень медленно), либо (что несколько лучше) используется таблица логарифмов.

Этот метод хорош для $m \leq 4$ поля $GF(2^m)$, но практически неприемлем при больших значениях m, особенно потому, что кроме таблицы логарифмов необходимо иметь также таблицу антилогарифмов такого же объема.

5) *Использование логарифмов Зеча* [12].

При этом способе для умножения используется только представление элементов поля в виде степеней примитивного элемента α конечного поля. Умножение выполняется как сложение по модулю $(2^m - 1)$ степеней примитивного элемента α, в виде которых представлены элементы поля. Сложение же выполняется с помощью так называемых логарифмов Зеча. В этом случае нет необходимости иметь таблицу антилогарифмов.

6) *Умножение в матричных вычислительных устройствах (МВУ)* [5].

МВУ позволяют на заданной элементной базе достичь высокого быстродействия, а однородность структуры делает возможным их реализацию в виде БИС.

Рассмотрим более подробно некоторые из способов выполнения операций над элементами конечного поля $GF(2^m)$, имеющими полиномиальное и векторное представления.

Если элементы поля $GF(p^m)$ представлять в виде многочленов над полем $GF(p)$, степень которого меньше m, то:

– сложение элементов a, $b \in GF(p^m)$ выполняется по правилу сложения представляющих их многочленов, т.е. $a(x)+b(x)=c(x)$;

– умножение элементов a, $b \in GF(p^m)$ выполняется по правилу умножения представляющих эти элементы многочленов по модулю заданного неприводимого многочлена $p(x)=x^m+p_{m-1}x^{m-1}+\ldots+p_0x^0$, т. е. $a(x)\cdot b(x)\equiv c(x) \bmod p(x)$;

– делению одного элемента $a \in GF(p^m)$ на другой элемент $b \in GF(p^m)$ соответствует умножение многочлена $a(x)$ на многочлен $c(x)$, соответствующий элементу $c \in GF(p^m)$, обратному $b \in GF(p^m)$, где многочлен $c(x)$ должен удовлетворять условию $b(x)\cdot c(x)\equiv 1 \bmod p(x)$.

4. Умножение элементов конечного поля GF(2^m)

4.1. Умножение элементов конечного поля на сдвиговых регистрах.

Определим сложности реализации операции умножения на сдвиговых регистрах для схемы, приведенной в [3] и содержащей 2 сдвиговых регистра, суммирующий регистр и блок логического умножения. Для хранения неприводимого многочлена $p(x)$ можно иметь регистр.

Из [9] известно, что простейшей схемой регистра сдвига является схема, построенная на *D*-триггерах. Простейшую схему суммирующего регистра можно построить на синхронизируемом двухтактном *T*-триггере [9]. Регистр для хранения многочлена $h(x)$ может быть построен на однотактных триггерах. Поэтому сложность реализации умножения элементов конечного поля будем определять для схемы, построенной с использованием этих триггеров.

Рассмотрим построение схем умножения элементов конечного поля на сдвиговых регистрах (ССУЭ) для любого неприводимого многочлена $p(x)$ заданной степени;

Так как подачу в ССУЭ значений величин p_i (0 или 1) полинома $h(x)$ в двоичном представлении $[p_{m-1}, p_{m-2}, \ldots, p_0]$ можно обеспечить без использования регистра (например, подавая на входы соответствующие уровни сигналов), то будем рассчитывать сложность для двух случаев: с наличием регистра для хранения многочлена $h(x)$ и без такового.

Время умножения будем определять с учетом времени, необходимого для предварительной записи в регистры перемножаемых элементов. Для рассматриваемой схемы перемножаемые элементы записываются в регистры параллельно. Для простоты расчетов примем, что оба сомножителя поступают в ССУЭ одновременно.

В результате анализа и расчетов получим следующие выражения для определения аппаратурной и временной сложности ССУЭ.

Аппаратурная сложность составляет:

а) без учета сложности регистра для хранения многочлена $h(x)$:

$$N^{И}_{ССУЭ}=77m-2;$$

б) с учетом сложности регистра для хранения многочлена $h(x)$:

$$\overline{N}^{И}_{ССУЭ}=86m-11.$$

Время умножения равно:

$$T^{И}_{ССУЭ}=(22m+15)t\ .$$

Отметим, что время умножения не зависит от наличия или отсутствия регистра для хранения многочлена $h(x)$.

4.2. Умножение элементов конечного поля по Барти-Шнайдеру.

В [4] показано, что при умножении элементов $a=(a_{m-1}, a_{m-2}, \ldots, a_1, a_0)$ и $b=(b_{m-1}, b_{m-2}, \ldots, b_1, b_0)$ каждая их m составляющих $c=(c_{m-1}, c_{m-2}, \ldots, c_1, c_0)$ определяется из соотношения:

$$c_i=[a_0, a_1, \ldots, a_{m-1}]\cdot[M_i]\cdot\begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{m-1}\end{bmatrix}, \quad (2)$$

где $M_i=\left[\alpha_{kl}^{(i)}\right]$ – квадратная матрица порядка m, элемент которой $\alpha_{kl}^{(i)}\in GF(2)$ является i-й координатой произведения базисных элементов поля $GF(2^m)$, $i=\overline{0, m-1}$.

Построим комбинационные схемы умножения элементов конечного поля $GF(2^m)$, реализующие соотношение (2). Такие схемы будем называть схемами умножения элементов конечного поля по Барти-Шнайдеру (БШУЭ).

На рис. 1 приведена схема одного разряда устройства умножения элементов поля $GF(2^m)$.

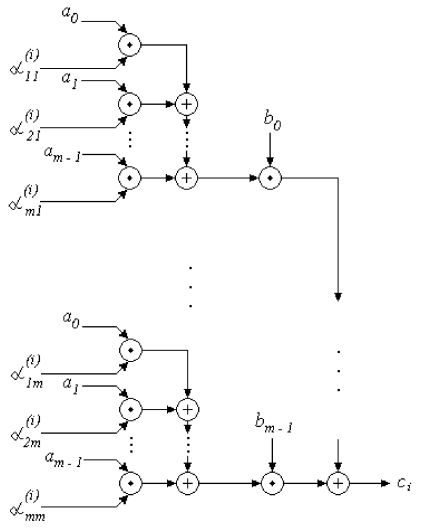

Рис. 1. Схема одного разряда устройства умножения элементов поля $GF(2^m)$

На рис. 2 показана универсальная схема (УБШУЭ) одного разряда устройства умножения элементов поля $GF(2^m)$, построенная на основе базовой ячейки (БЯ), состоящей из элемента *И* на два входа и двухвходового сумматора по модулю 2 и имеющей 3 входа и 1 выход.

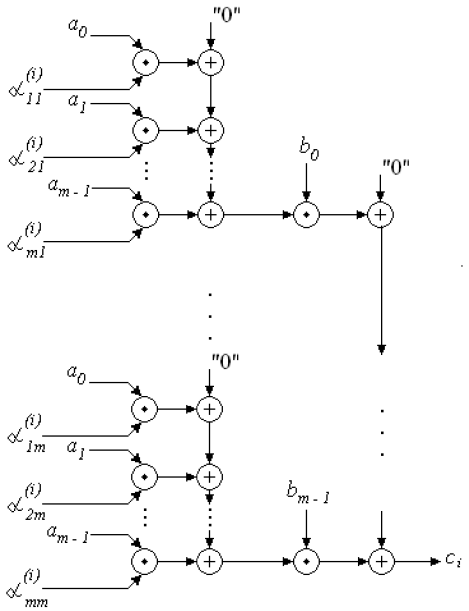

Рис. 2. Универсальная схема одного разряда устройства умножения элементов поля $GF(2^m)$ на БЯ (УБШУЭ)

Базовую ячейку можно составить из $(m+1)$ элементов *И* на два входа и $(m+1)$ двухвходовых сумматоров по модулю 2. Тогда схема одного разряда устройства умножения будет состоять из m таких базовых ячеек ($БЯ_{max}$). Каждая базовая ячейка будет иметь $(2m+3)$ входов и 1 выход.

Определим аппаратурную и временную сложности схем умножения для любого неприводимого многочлена $p(x)$ заданной степени с предварительным вычислением величин $\alpha_{kl}^{(i)}$ и их хранением.

Анализ и расчет показывает, что схема умножения по Барти-Шнайдеру (построенная на основе схемы одного разряда – рис. 1) имеет аппаратурную сложность (для вычисления всех величин c_i и без учета сложности вычисления величин $\alpha_{kl}^{(i)}$):

$$N_{БШУЭ}^{И}=(m+1)(5m-4)m.$$

Время умножения для этого случая составляет (при одновременном вычислении величин c_i и с учетом времени вычисления на схемах *И* величин $a_j \cdot \alpha_{kl}^{(i)}$):

$$T_{БШУЭ}^{И} = (6m-4)t\,.$$

Если схемы умножения (рис. 2) строятся на основе БЯ, то имеем:

$$N_{УБШУЭ} = 5m^2(m+1)\,,$$

$$T_{УБШУЭ} = (6m+2)t\,.$$

Кроме того, для обеих схем (рис. 1 и рис. 2) необходимо иметь m^3 ячеек памяти для хранения величин $\alpha_{kl}^{(i)}$.

Следует отметить, что схемы УБШУЭ можно адаптировать к изменению степени *m* многочлена $p(x)$.

4.3. Умножение элементов конечного поля на матричных вычислительных устройствах.

Схемы матричных устройств умножения элементов (МУУЭ) конечного поля $GF(2^m)$ [5] строятся на основе базовой ячейки (БЯ), состоящей из двух элементов *И* на два входа каждый и двух двухвходовых сумматоров по модулю 2.

Сложность МУУЭ составляет [5]:

$$N_{МУУЭ} = 10m^2\,,\quad T_{МУУЭ} = 7mt\,.$$

5. Инвертирование элементов конечного поля GF(2^m)

Известно, что в любом поле для любого ненулевого элемента существует обратный элемент по умножению, т. е. такой элемент, произведение которого на исходный элемент поля равно единице.

Пусть $a(x) = a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \ldots + a_1x + a_0$ – многочлен, представляющий элемент поля $GF(2^m)$, инверсное значение которого необходимо найти. Инверсией этого многочлена будет многочлен $b(x) = b_{m-1}x^{m-1} + b_{m-2}x^{m-2} + \ldots + b_1x + b_0$ такой, что $\left\langle a(x)\cdot b(x)\right|_{p(x)} = 1$ или:

$$a(x)\cdot b(x) = p(x)\cdot q(x) + c(x)\,, \qquad (3)$$

где $p(x) = p_mx^m + p_{m-1}x^{m-1} + \ldots + p_1x + p_0$;

$q(x) = q_{m-2}x^{m-2} + q_{m-3}x^{m-3} + \ldots + q_1x + q_0$;

$c(x) = c_{m-1}x^{m-1} + c_{m-2}x^{m-2} + \ldots + c_1x + c_0$

(здесь $c_{m-1} = c_{m-2} = \ldots = c_1 = 0\,,\quad c_0 = 1$).

Решение $b(x)$ может быть найдено по алгоритму Евклида. Но этот процесс сложно реализовать аппаратно, так как он имеет большое число промежуточных результатов, которые должны запоминаться.

Инвертирование элементов конечного поля, имеющих полиномиальное и векторное представления, может быть выполнено на регистрах сдвига [1 -3, 7], комбинационных схемах по способу Барти-Шнайдера [4], а также с помощью последовательности мультипликаторов, дающих возможность получения (2^m-2)-ой степени инвертируемого элемента β ($\beta^{-1} = \beta^{2^m-2}$), так как ненулевые элементы поля $GF(2^m)$ образуют циклическую мультипликативную группу порядка (2^m-1).

5.1. Инвертирование элементов конечного поля на сдвиговых регистрах.

Для нахождения многочлена $b(x)$, удовлетворяющего (3), Берлекэмпом [7] на основе алгоритма деления Евклида предложена процедура вычисления с использованием двух основных регистров сдвига, а также требующая дополнительно наличие маркерного регистра и схемы управления [3].

При построении схем (устройств) инвертирования элементов конечного поля на сдвиговых регистрах (ССИЭ) можно использовать сумматор по модулю 2, *D*-триггер и *T*-триггер, которые соответственно реализуются с помощью 4, 18 и 21 элементов базисного набора. Тогда в результате расчёта получаем, что аппаратурная сложность ССИЭ, приведенной в [3], для любого неприводимого многочлена $p(x)$ заданной степени равна:

$$N_{ССИЭ}^{И} = 102(m+2) + 64\,.$$

Для наиболее простой реализации инвертирования с помощью процедуры Берлекэмпа требуется до (2m+1) тактов [3].

5.2. Инвертирование элементов конечного поля с помощью последовательности мультипликаторов.

Существует много путей получения (2^m-2)-ой степени инвертируемого элемента β. Эффективным способом является формирование последовательности $\beta, \beta^2, \beta^3, \ldots, \beta^{2^m-2}$ путём использования соотношения

$$\beta^2 \cdot \beta^4 \cdot \beta^8 \cdot \ldots \cdot \beta^{2^m-1} = \\ = \{[(\beta^2)\cdot\beta]^2 \cdot \beta \ldots\}^2 = \beta^{2^m-2}.$$

Каждый шаг процедуры требует перемножения текущей последовательности на β и возведения в квадрат результата. Всего в процедуре $(m-1)$ шаг.

Инверсия (относительно умножения) с помощью последовательности мультипликаторов может формироваться, если допускает время (например, при малом количестве необходимых делений).

6. Новые методы вычислений в конечных полях

В работах [13 - 16] предложены методы, позволяющие выполнять:

1) операции над многочленами в поле *GF*(2):

– умножение многочленов [13],

– умножение на фиксированный многочлен[13],

– деление многочленов [14];

2) умножение элементов конечного поля $GF(2^m)$ [15, 16].

Определены также сложности комбинационных схем, реализующих эти операции.

Для комбинационных схем умножения многочленов (КСУМ):

$N_{КСУМ} = 5mn - 4(m+n) + 4$ – для $m<n$ и $m>n$,

$T_{КСУМ} = (3m-2)t$ – для $m<n$,

$T_{КСУМ} = (3n-2)t$ – для $m>n$.

Для универсальных комбинационных схем умножения многочленов (УКСУМ):

$N_{УКСУМ} = 5mn$ – для $m<n$ и $m>n$,

$T_{УКСУМ} = (3m+1)t$ – для $m<n$,

$T_{УКСУМ} = (3n+1)t$ – для $m>n$.

Для комбинационных схем деления многочленов (КСДМ):

$$N_{КСДМ} = 5(m-1)n = 5(m-1)(d-m+1),$$

$$T_{КСДМ} = 4(d-m+1)t.$$

Для комбинационных схем умножения элементов конечного поля (КСУЭ и УКСУЭ) [16]:

$$N_{КСУЭ} = m(10m-13)+4,$$

$$N_{УКСУЭ} = 5m(2m-1),$$

$$T_{КСУЭ} = (6m-2)t,$$

$$T_{УКСУЭ} = (6m-8)t.$$

Дадим сравнительную оценку аппаратурной и временной сложностей схем, реализующих методы выполнения операций над многочленами в поле *GF*(2) и элементами поля $GF(2^m)$, предложенные в [13 - 16], и аналогичных параметров известных схем, построенных в [1 - 5].

В результате сравнения КСУМ и ССУМ (при наличии регистра для записи и хранения сомножителя в ССУМ) получаем:

1) при $\deg a(x) = \deg b(x)$:

$$N_{КСУМ} - N_{ССУМ} = 5m^2 - 42m + 27,$$

$$T_{ССУМ} - T_{КСУМ} = (37m-27)t,$$

т. е. КСУМ лучше ССУМ:

– по аппаратурной сложности – для $m < 8$,

– по временной сложности – для $m \geq 1$;

2) при $\deg a(x) \neq \deg b(x)$:

$$N_{ССУМ} - N_{КСУМ} = 38n - 27 - m(5n-4)$$

– для $n > m$ и $m > n$,

$$T_{ССУМ} - T_{КСУМ} = (17m + 20n - 31)t$$

– для $n > m$,

$$T_{ССУМ} - T_{КСУМ} = (17n + 20m - 31)t$$

– для $m > n$,

т. е. КСУМ лучше ССУМ:

– по аппаратурной сложности – для $n < (4m - 27)/(5m - 38)$,

– по временной сложности – для любых m и n.

Сравнение КСУМ и МУУМ показывает:

1) при $\deg a(x) = \deg b(x)$ сложность КСУМ меньше сложности МУУМ:

$$N_{МУУМ} - N_{КСУМ} = 4(2m - 1),$$

$$T_{МУУМ} - T_{КСУМ} = 3t;$$

2) при $\deg a(x) \neq \deg b(x)$ сложность КСУМ меньше сложности МУУМ:

$N_{МУУМ} - N_{КСУМ} = 4(m + n - 1)$ – для $n > m$ и $m > n$,

$T_{МУУМ} - T_{КСУМ} = 3t$ – для $m > n$,

$T_{МУУМ} - T_{КСУМ} = 3(n - m + 1)t$ – для $n > m$.

Сравнение УКСУМ и МУУМ дает следующее:

1) при $\deg a(x) = \deg b(x)$ сложности одинаковы;

2) при $\deg a(x) \neq \deg b(x)$:

– аппаратурная сложность одинаковая,

– время умножения одинаковое при $m > n$,

– время умножения на МУУМ больше на величину $3(n - m)t$ при $n > m$.

Поэтому из-за меньшей временной сложности при $n > m$ более простой является УКСУМ.

Сравнивая КСДМ и ССДМ (при наличии регистра для записи и хранения делителя в ССДМ) получаем:

$$N_{КСУМ} - N_{ССУМ} = (m - 1)[5(d - m + 1) - 34] - 1,$$

$$T_{ССУМ} - T_{КСУМ} = (16d + 4m + 3)t,$$

т. е. КСДМ лучше ССДМ:

– по аппаратурной сложности – для $(d - 1) - (m - 1) \leq 5$,

– по временной сложности – для $d \geq 1$ и $m \geq 1$.

Для КСДМ и МУДМ сложность одинаковая. Аппаратурная сложность больше для МУДМ на величину $5(d - m + 1)$.

Аппаратурная сложность КСУЭ меньше аппаратурной сложности ССУЭ для $m \leq 9$. Временная сложность меньше для КСУЭ на величину $(16m + 17)$.

Сложность КСУЭ и УКСУЭ, реализующих метод умножения элементов конечного поля $GF(2^m)$, предложенный в [16], меньше сложности матричных вычислительных устройств [5] и схем умножения по способу, предложенному в [4].

Выводы

Проведенный анализ методов вычислений в конечных полях и сравнительная оценка сложности схем, реализующих эти методы, показал, что разработанные в [13 - 16] методы умножения и деления многочленов над полем $GF(2)$ и умножения элементов конечного поля $GF(2^m)$ имеют преимущество по сравнению с ранее известными методами. Кроме того, эти методы позволяют создавать быстродействующие схемы, структура которых однородна и универсальна, в отличие от структуры линейных последовательностных машин, что делает их перспективными для реализации в виде БИС.

Полученные математические выражения для определения сложности схем, реализующих вычисления с помощью регистров сдвига, а также схем, построенных по методу Барти-Шнайдера, позволили сделать сравнения этих схем со схемами, реализующими новые методы.

Список литературы

1. *Питерсон У., Уэлдон Э.* Коды, исправляющие ошибки: Пер. с англ. / Под ред. Р.Л. Добрушина и С.И. Самойленко. – М.: Мир, 1976. – 596 с.

2. *Гилл А.* Линейные последовательностные машины. – М.: Наука, 1974. – 287 с.

3. *Блох Э. Л., Зяблов В. В.* Обобщенные каскадные коды. – М.: Связь, 1976. – 240 с.

4. *Bartee T.C., Shneider D.I.* Computations with finite fields. – Information and control, 1963. – 6. – P. 79–98.

5. *Смолов В. Б., Шумилов Л. А., Зайкова Л. А.* Построение матричных вычислительных устройств для выполнения операций над многочленами и элементами конечных полей GF(2^m). – Электронное моделирование, 1979. – №2. – С. 63–67.

6. *Coppersmith D.* – IEEE Trans., 1984, v. IT-30, №4.

7. *Берлекэмп Э.* Алгебраическая теория кодирования. – М.: Мир, 1971. – 477 с.

8. *Блейхут Р.* Теория и практика кодов, контролирующих ошибки: Пер. с англ. / Под ред. К. Ш. Зигангирова. – М.: Мир, 1986. – 576 с.

9. *Каган Б. М.* Электронные вычислительные машины и системы. – М.: Энергия, 1979. – 528 с.

10. *Golomb S.W.* Shift Register Sequences. – Holden-Day. San Francisco, 1967.

11. *Kautz W.H., ed.* Linear sequential switching circuits: Selected papers. – Holden-Day. San Francisco, 1965.

12. *Conway J. H.* A tabulation of some information concerning finite fields, in: R.F. Churchhouse and J.-C. Herz eds., Computers in Mathematical Research, (North-Holland, Amsterdam, 1968) – P. 37–50.

13. *Кубицкий В.И.* Операции над многочленами в поле GF(2). – Научный вестник ГосНИИ “Аэронавигация”, серия «Проблемы организации воздушного движения. Безопасность полетов». №7. – М.: 2007. – С. 185–194.

14. *Кубицкий В.И.* Деление многочленов над полем GF(2). Научный Вестник МГТУ ГА, № 132 (8). – М.: МГТУ ГА, 2008. – С. 86–93.

15. *Жуков И.А., Кубицкий В.И., Дрововозов В.И.* Алгоритмы выполнения операций над элементами конечного поля GF(2^m) в вычислительных устройствах. – Матеріали VIII Міжнародноі науково-технічноі конференціі “ABIA-2007”. – Т.1. – К.: НАУ, 2007. – С. 13.5–13.8.

16. *Кубицкий В.И.* Умножение элементов конечного поля GF(2^m). – Научный Вестник МГТУ ГА, № 145 (8). – М.: МГТУ ГА, 2009. – С. 105–112.