УДК 681.3.001.2

И.В. ГОРМАКОВА, ассистент НТУ «ХПИ»;
Р.М. АЛИЕВ, магистр НТУ «ХПИ»

МЕТОД СИНТЕЗА УМНОЖИТЕЛЕЙ МОНТГОМЕРИ В ПОЛЯХ ГАЛУА С БЛОЧНО-МОДУЛЬНОЙ АРХИТЕКТУРОЙ

В статье описывается новый метод построения пословно-последовательного умножителя Монтгомери, который базируется на представлении элементов поля GF(2p) в стандартном базисе. Полученный умножитель имеет каскадную архитектуру, которая легко тестируется. Предложенный умножитель может быть с легкостью построен для любого поля GF(2p) и для любого генерирующего полинома F(x).

У статті описується новий метод побудови послівно-послідовного помножувача Монтгомері, який базується на поданні елементів поля $GF(2^p)$ у стандартному базисі. Отриманий помножувач має каскадну архітектуру, що легко тестується. Запропонований помножувач може бути з легкістю побудований для будь-якого поля $GF(2^p)$ та для будь-якого генеруючого полінома F(x).

In this paper a new word-serial Montgomery multiplier in $GF(2^p)$ for standard-basis representation is developed. Obtained multiplier architecture is scaleable and easy-to-test. Proposed multiplier can be easily designed for any field $GF(2^p)$ and any field-generator polynomial F(x).

Постановка проблемы. В настоящее время приоритетным направлением в области приборостроения и схемотехники является разработка компактных высокоскоростных схем, способных работать с многоразрядными данными. Кроме вышеперечисленный свойств, разрабатываемые модули и устройства должны отвечать требованием тестопригодности и отказоустойчивости.

При проектировании систем управления железнодорожным транспортом, банковской деятельностью, объектов АСУ ТП и т.п. обеспечение и повышение безопасности таких систем является одним из главных требований. Наилучшим решением для безопасной передачи и хранения информации является применение систем защиты информации, среди которых наиболее часто используемыми являются криптосистемы

Известно, что в криптосистемах широко используются арифметические модули, функционирующие в полях Галуа $GF(2^p)$ [1]. Среди арифметических операций, проводимых над элементами конечного поля, наиболее важными и часто используемыми являются операции умножения и возведения в квадрат. Другие арифметические операции, такие как инверсия и возведение в степень, могут быть выражены через операции умножения и возведения в квадрат.

Для быстрого умножения многоразрядных чисел в конечных полях был предложен алгоритм умножения Монтгомери [2]. В настоящее время умножители Монтгомери находят широкое применения при построении крипто-

графических процессоров, реализующих криптоалгоритмы в эллиптических кривых [3].

Анализ литературы. В [4] представлены архитектуры параллельного и разрядно-последовательно умножителя Монтгомери в поле $GF(2^p)$. Разрядно-последовательные умножители имеют наиболее простую архитектуру, однако время выполнения операции умножения в поле $GF(2^p)$ составляет p тактов. В параллельных умножителях операция умножения выполняется за один такт, однако аппаратные затраты и площадь на кристалле достаточно велики. В [5] представлена архитектура пословно-последовательного умножителя Монтгомери. Показано, что такие умножители наилучшим образом соответствует требованиям временных (время выполнения алгоритма умножения), аппаратных (количество логических вентилей) и пространственных (площадь, занимаемая на кристалле) затрат.

Целью статьи является разработка метода синтеза умножителя Монтгомери в поле Галуа $GF(2^p)$ с блочно-модульной архитектурой, выполняющего операцию пословно-последовательного умножения.

Предложенный в настоящей статье метод синтеза основан на подходе к построению пословно-последовательных умножителей из [6].

В предлагаемой архитектуре умножителя выполняется операция умножения по модулю неприводимого полинома, используя так называемый стандартный базис представления элементов поля $GF(2^p)$.

Конечное поле $GF(2^p)$ всегда связано с некоторым неприводимым полиномом степени p, который является образующим полиномом поля:

$$F(x)=x^p+f_{p-1}x^{p-1}+f_{p-2}x^{p-2}+\ldots+f_1x+1, \quad f_i\in GF(2) \qquad (1)$$

Пусть элемент α является корнем неприводимого полинома $F(x)$, удовлетворяющим условию $F(\alpha)=0$, следовательно, элемент α образует все ненулевые элементы поля $\{\alpha, \alpha^2, \ldots, \alpha^{2^p-1}\}$. Элемент $\alpha\in GF(2^p)$ называется образующим элементом поля. Произвольный элемент поля может быть задан как полином степени $(p\text{-}1)$ над полем $GF(2)$, то есть

$$B=b_0+b_1\alpha+b_2\alpha^2+\ldots+b_{p-1}\alpha^{p-1} \text{ при } b_i\in GF(2) \qquad (2)$$

Рассмотрим процедуру умножения элементов поля $GF(2^p)$ по методу Монтгомери. Пусть даны два элемента поля $GF(2^p)$ A' и B', представленные в стандартном базисе, для которых $\varphi=A'\cdot B' \bmod F(\alpha)$ – произведение элементов A' и B' в поле $GF(2^p)$. Пусть A и B – $F(\alpha)$-вычеты, определенные как

$$A = A'\cdot R \bmod F(\alpha) = \sum_{i=0}^{p-1} a_i\alpha^i,\ a_i\in GF(2) \qquad (3)$$

и

$$B = B'\cdot R \bmod F(\alpha) = \sum_{i=0}^{p-1} b_i\alpha^i,\ b_i\in GF(2) \qquad (4)$$

где R – одночлен, называемый фактор-множителем Монтгомери, который удовлетворяет условию НОД(R, $F(\alpha)$)=1.

В умножителе Монтгомери произведение C операндов A и B, являющихся элементами поля $GF(2^p)$, представлено в виде

$$C=AB\,R^{-1}\bmod F(\alpha) \qquad (5)$$

где R – фактор-множитель Монтгомери и $R=\alpha^p$.

В общем случае алгоритм умножения Монтгомери можно записать следующим образом:

Входные данные: $A, B \in GF(2^p)$, $F(\alpha)$

Выходные данные: $C = A\cdot B\cdot \alpha^{-p} \bmod F(\alpha)$

ШАГ 1: C=0

ШАГ 2: Для i=0 до (p-1)

ШАГ 3: $C=C+a_iB$

ШАГ 4: $C=C+c_0F(\alpha)$

ШАГ 5: $C=C/\alpha$

При пословно-последовательном умножении элементов поля один из операндов разбивается на слова. Разделим операнд A на $\lceil p/\omega \rceil = k$ слов длинной в ω бит. Тогда операнд A может быть представлен в виде полинома:

$$A=A_{k-1}\alpha^{(k-1)\omega}+\ldots+A_2\alpha^{2\omega}+A_1\alpha^{\omega}+A_0 \qquad (6)$$

где A_j – полином степени $\le(\omega-1)$, j=0,…, (k-1). Причем степень полинома A_{k-1} может быть меньше чем (ω-1).

Каждое полученное слово в свою очередь также может быть представлено в виде полинома:

$$A_j=a_{\omega j+(\omega-1)}\alpha^{(\omega-1)}+a_{\omega j+(\omega-2)}\alpha^{(\omega-2)}+\ldots+a_{\omega j+2}\alpha^2+a_{\omega j+1}\alpha+a_{\omega j},\ j=0,\ldots,(k-1) \qquad (7)$$

Подставляем в выражение (5) выражение (6), получим:

$$\begin{aligned} C &= (A_{k-1}\alpha^{(k-1)\omega} + A_{k-2}\alpha^{(k-2)\omega} + \ldots + A_2\alpha^{2\omega} + \\ &+ A_1\alpha^{\omega} + A_0)BR^{-1} \bmod F(\alpha) = \\ &= (A_{k-1}B\alpha^{(k-1)\omega}\alpha^{-u} + A_{k-2}B\alpha^{(k-2)\omega}\alpha^{-u} + \ldots + \\ &+ A_2B\alpha^{2\omega}\alpha^{-u} + + A_1B\alpha^{\omega}\alpha^{-u} + A_0B\alpha^{-u}) \bmod F(\alpha) = \\ &= (A_{k-1}B\alpha^{(k\omega-u)}\alpha^{-\omega} + A_{k-2}B\alpha^{(k\omega-u)}\alpha^{-2\omega} + \ldots + \\ &+ A_2B\alpha^{(k\omega-u)}\alpha^{-(k-2)\omega} + A_1B\alpha^{(k\omega-u)}\alpha^{-(k-1)\omega} + \\ &+ A_0B\alpha^{(k\omega-u)}\alpha^{-k\omega}) \bmod F(\alpha) \end{aligned} \qquad (8)$$

В выражении (8) вынесем за скобку общий множитель $\alpha^{-\omega}$. Получим:

$$\begin{aligned} C &= (A_0B\alpha^{(k\omega-u)}\alpha^{-(k-1)\omega} + A_1B\alpha^{(k\omega-u)}\alpha^{-(k-2)\omega} + \\ &+ A_2B\alpha^{(k\omega-u)}\alpha^{-(k-3)\omega} + \ldots + \\ &+ A_{k-2}B\alpha^{(k\omega-u)}\alpha^{-\omega} + A_{k-1}B\alpha^{(k\omega-u)})\alpha^{-\omega} \bmod F(\alpha) \end{aligned} \qquad (9)$$

В полученной в скобках сумме также возможно вынести за скобки общий множитель $\alpha^{-\omega}$. Процедура вынесения за скобки общего множителя продолжается до тех пор, пока степень α при A_0 не станет равной –ω. Кроме того, введем обозначение $B^{(0)}=B\cdot\alpha^{(k\omega-p)}$:

$$C=[((\ldots(A_0B^{(0)}\alpha^{-\omega}+A_1B^{(0)})\alpha^{-\omega}+\ldots+A_{k-2}B^{(0)})\alpha^{-\omega}+ \\ +A_{k-1}B^{(0)})\alpha^{-\omega}]\bmod F(\alpha) \qquad (10)$$

На основании выражения (10) сформулируем алгоритм пословно-последовательного умножения Монтгомери в поле $GF(2^p)$.

Входные данные: A, $B\in GF(2^p)$, $F(\alpha)$

Выходные данные: $C=A\cdot B\cdot\alpha^{-u}\bmod F(\alpha)$

ШАГ 1: Вычислить значение $B^{(0)}=B\cdot\alpha^{(k\omega-u)}\bmod F(\alpha)$

ШАГ 2: Установить $C_{-1}=0$

ШАГ 3: Установить счетчик $i=0$. Для $[i=0\div(k-1)]$ повторить следующую последовательность действий:

ШАГ 4: Вычислить значение $D_i=A_i\,B^{(0)}\bmod F(\alpha)$

ШАГ 5: Вычислить значение $C_i=C_{i-1}\cdot\alpha^{-\omega}+D_i$

ШАГ 6 Увеличить значение счетчика i на 1. Если $i<k$, перейти к шагу 4, иначе перейти к шагу 7

ШАГ 7: Вычислить значение $C=C_{(k-1)}\cdot\alpha^{-\omega}\bmod F(\alpha)$

ШАГ 8: Конец алгоритма.

На основании приведенного выше алгоритма разработана структурная схема пословно-последовательного умножителя Монтгомери, представленная на рисунке 1.

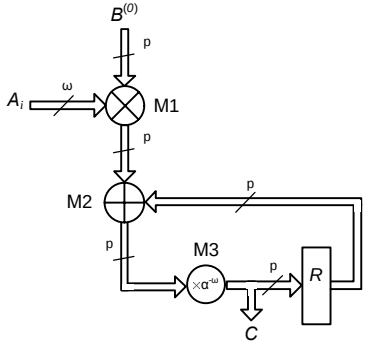

Рис. 1. Структурная схема пословно-последовательного умножителя Монтгомери

На первом шаге алгоритма вычисляется значение $B^{(0)}=B\cdot\alpha^{(k\omega-p)}\bmod F(\alpha)$. При $(p/\omega)=k$ и $R=\alpha^p$ значение $B^{(0)}=B\cdot\alpha^0\bmod F(\alpha)=B$. В противном случае $B^{(0)}=B\cdot\alpha^n\bmod F(\alpha)$, где $n=k\omega-p$ и $1\le n\le(\omega-1)$

На втором шаге выполняется операция $C_{-1}=0$, что соответствует обнулению регистра R (см. рис.1). Далее для каждого слова A_i, $i=0\ldots(k-1)$ на шаге 4-5 выполняется вычисление частичного произведения. На шаге 4 выполняется умножение по модулю $F(\alpha)$ текущего слова A_i на операнд $B^{(0)}$:

$$D_i=A_i\,B^{(0)}=a_0B^{(0)}+a_1B^{(0)}\alpha+a_2B^{(0)}\alpha^2+\ldots+a_{\omega-1}B^{(0)}\alpha^{\omega-1} \qquad (11)$$

Вычисление произведения $A_iB^{(0)}\mathrm{mod}F(\alpha)$ выполняется в блоке М1 (рис.1).

В блоке М2 происходит сложение полученного результата с содержимым регистра R. Блок М3 выполняет умножение полученной суммы C_i на $\cdot\alpha^{-\omega}$ по модулю $F(\alpha)$ с последующим сохранением результата в регистре R. После последовательной обработки всех слов на шаге 7 происходит умножение $C_{(k-1)}$ на $\alpha^{-\omega}$ по модулю $F(\alpha)$. Результат умножения C снимается с выхода блока М3.

На основании приведенного алгоритма была разработана блочно-модульная архитектура умножителя, реализующего алгоритм пословно-последовательного умножения Монтгомери (рис.2).

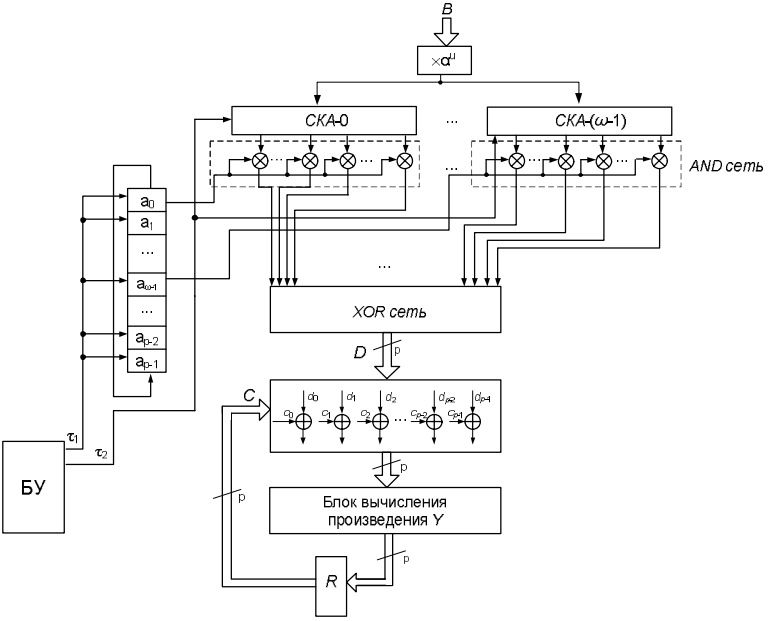

Рис. 2. Блочно-модульная архитектура пословно-последовательного умножителя Монтгомери в конечных полях

В состав пословно-последовательного умножителя входят следующие блоки:

1) блок вычисление произведения $B^{(0)}=B\cdot\alpha^{\mu}\mathrm{mod}F(\alpha)$;

2) СКА – сети клеточных автоматов;

3) сдвиговый регистр с первым операндом A;

4) AND сети;

5) XOR сеть;

6) блок суммирования промежуточных результатов D_i и C_i;

7) блок вычисления произведения $Y=L\cdot\alpha^{-\omega} \bmod F(\alpha)$;

8) регистр R.

Модуль вычисление произведения $B^{(0)}=B\cdot\alpha^{n} \bmod F(\alpha)$ представляет собой специальным образом построенную сеть из XOR вентилей.

Блок М1 структурной схемы рис.1 реализован с помощью СКА, AND сетей и XOR сети. СКА предназначены для последовательного вычисления произведений $B^{(0)}$, $B^{(0)}\alpha$, …, $B^{(0)}\alpha^{\omega-1}$ за ω тактов. После такта ω функционирование СКА прекращается. Вычисленные значения $B^{(0)}$, $B^{(0)}\alpha$, …, $B^{(0)}\alpha^{\omega-1}$ хранятся соответственно в СКА-0, СКА-1, …, СКА-(ω-1). Структура СКА аналогична структуре, приведенной в [6].

Каждая из ω AND сетей состоит из p двухвходовых вентилей AND. Один из входов i-го AND вентиля m-ой AND сети запитан выходом i-ой ячейки m-ой СКА. На второй вход всех вентилей m-ой AND сети по общей одноразрядной шине подается один бит a_m из входного слова A_i.

XOR сеть предназначена для последовательного суммирования произведений $a_0B^{(0)}$, $a_1B^{(0)}\alpha$, $a_2B^{(0)}\alpha^2$,…, $a_{\omega-1}B^{(0)}\alpha^{\omega-1}$. На выходе XOR сети формируется частичное произведение $D=A_iB^{(0)} \bmod F(\alpha)$.

Блок М2 структурной схемы рис.1 реализован с помощью модуля суммирования промежуточных результатов, состоящего из p двухвходовых XOR вентилей и предназначенного для побитового суммирования двух операндов $L=D\oplus C$, где первый операнд D – частичное произведение, второй операнд C – содержимое регистра R.

Блок М3 структурной схемы рис.1 реализован с помощью модуля вычисления произведения Y, который представляет собой специальным образом построенную сеть из XOR вентилей, обеспечивающую вычисление произведения $Y=L\cdot\alpha^{-\omega} \bmod F(\alpha)$. Структурная схема модуля вычисления произведения Y представлена на рисунке 3.

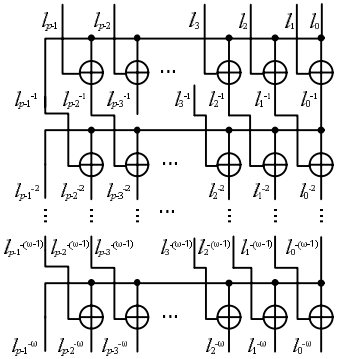

Рис. 3. Структурная схема блока вычисления произведения Y

В представленной схеме верхние индексы битов обозначают степень образующего элемента α, на который умножается элемент поля $L=[l_{p-1}, l_{p-2}, \ldots, l_2, l_1, l_0]$. Наличие вентилей XOR в схеме в определенной позиции определяется образующим полиномом поля $F(x)$. Если коэффициент f_i образующего полинома $F(x)$ равен 1, то $l_{i-1}^{-j} = l_0^{-(j-1)} \oplus l_i^{-(j-1)}$ для $j=1, \ldots, \omega$. При $f_i=0$ $l_{i-1}^{-j} = l_i^{-(j-1)}$.

Рассмотри алгоритм работы умножителя.. На нулевом такте операнд B поступает на вход модуля вычисление произведения $B^{(0)}=B\cdot\alpha^n \bmod F(\alpha)$. Далее в СКА-0, СКА-1, …, СКА-(ω-1) загружается операнд $B^{(0)}$, регистр R обнуляется. На первом такте на выходах всех СКА формируется значение $B^{(0)}$. Функционирование СКА-0 после первого такта прекращается. Таким образом, состояние СКА-0 далее остается неизменным и равно $B^{(0)}$. На втором такте на выходах СКА-1, …, СКА-(ω-1) формируется значение $B^{(0)}\alpha$ и прекращается функционирование СКА-1. На такте ω на выходах СКА-0, СКА-1, …, СКА-(ω-1) будут соответственно значения $B^{(0)}$, $B^{(0)}\alpha$, …, $B^{(0)}\alpha^{\omega-1}$, которые сохраняются до окончания выполнения операции умножения двух элементов поля. После каждого такта выходные значения СКА поступают на входы AND сетей, которые последовательно вычисляют произведения $a_iB^{(0)}\alpha^j$ для текущего слова A_0. Выходы AND сетей заводятся на XOR сеть.

На такте ω в XOR сети происходит вычисление суммы D всех произведений $a_0B^{(0)}$, $a_1B^{(0)}\alpha$, $a_2B^{(0)}\alpha^2$,…, $a_{\omega-1}B^{(0)}\alpha^{\omega-1}$. Далее значение D поступает на модуль суммирования промежуточных результатов, в котором происходит суммирование D с содержимым регистра R. Так как содержимое регистра равно нулю, то на выходе модуля суммирования промежуточных результатов получаем значение $L=A_0B^{(0)}$. Далее значение L поступает на вход модуля вычисления произведения $Y=L\cdot\alpha^{-\omega} \bmod F(\alpha)$. Выходным значением модуля на такте ω будет значение $Y=A_0B^{(0)}\alpha^{-\omega} \bmod F(\alpha)$, которое записывается в регистр R.

На такте (ω+1) происходит циклический сдвиг регистра с первым операндом A вправо на ω разрядов. Следовательно, на входы AND сетей поступают биты слова A_1. Далее для слова A_1 выполняется та же последовательность действий, что и для слова A_0. Таким образом, на выходе модуля суммирования промежуточных результатов получаем значение $L=A_0B^{(0)}\cdot\alpha^{-\omega} \bmod F(\alpha)+A_1B$. Выходным значением модуля вычисления произведения Y на такте (ω+1) будет значение $Y=(A_0B^{(0)}\cdot\alpha^{-\omega} \bmod F(\alpha)+A_1B^{(0)})\cdot\alpha^{-\omega} \bmod F(\alpha)$, которое записывается в регистр R. На протяжении последующих тактов выполняются аналогичные операции для слов A_2, .., A_{k-1}.

Общее время работы умножителя составляет ($\lceil p/\omega \rceil+\omega$) тактов. На ($\lceil p/\omega \rceil+\omega$) такте с выхода модуля вычисления произведения Y снимается значение произведения, вычисленное по формуле (10).

В предложенной архитектуре пословно-последовательного умножителя Монтгомери в поле $GF(2^p)$ используются унифицированные блоки из сетей клеточных автоматов, комбинационных модулей и регистров, что позволяет легко модифицировать архитектуру умножителя при изменении длины операндов, длины слова, образующего полинома поля и просто реализовать умножитель на ПЛИС типа FPGA. Изменение образующего полинома при сохранении степени полинома p требует лишь изменения правил настройки сети клеточных автоматов, входящих в состав умножителя, при полном сохранении их структуры.

Выводы. Предложенный метод синтеза умножителя Монтгомери в поле Галуа $GF(2^p)$, выполняющего операцию пословно-последовательного умножения, позволяет синтезировать умножитель с блочно-модульной архитектурой, которая соответствует требованиям быстродействия, каскадности и тестопригодности.

Список литературы: 1. *N. Petra, D. De Caro and A.G.M. Strollo*. A novel architecture for Galois Fields $GF(2^m)$ multipliers based on Mastrovito scheme. IEEE Trans.Comput., 2007, Nov., vol. 56, pp.1470-1483. 2. *P. L. Montgomery*. Modular multiplication without trial division. // Mathematics of Computation, 1985. vol. 44, pp. 519-521. 3. *G. Orlando, C. Paar*. A high performance reconfigurable elliptic curve processor for $GF(2^m)$. Proc. Second Int'l Workshop Cryptographic Hardware and Embedded Systems (CHES '00), K. Koc and C. Paar, eds., pp. 41-56, 2000. 4. *Arash Hariri, Arash Reyhani-Masoleh*. Bit-serial and bit-parallel Montgomery multiplication and squaring over $GF(2^m)$. IEEE Trans.Comput., 2009, Oct., vol. 58, pp.1332-1345. 5. *E. Savaş, A. F. Tenca, and Ç. K. Koç*. A scalable and unified multiplier architecture for Finite Fields $GF(p)$ and $GF(2^m)$. Proc. Second Int'l Workshop Cryptographic Hardware and Embedded Systems – CHES 2000, Ç. K. Koç and C. Paar, eds., pp. 277-292, Aug. 2000. 6. *Дербунович Л.В., Гормакова И.В.* Методы построения арифметических модулей, оперирующих в полях Галуа. // Вестник НТУ «ХПИ», №23, 2010 г., стр. 34-39.

Статья представлена д.т.н. проф. НТУ «ХПИ» Дербуновичем Л.В.
Поступила в редколлегию 04.04.12