МЕТОД БЫСТРОГО ВЫЧИСЛЕНИЯ ПРЕОБРАЗОВАНИЯ ФУРЬЕ НАД КОНЕЧНЫМ ПОЛЕМ

П.В. Трифонов, С.В. Федоренко

15 июля 2003 г.

Аннотация

Рассматривается задача быстрого вычисления преобразования Фурье над конечным полем путем разложения произвольного многочлена на сумму линеаризованных. Приводятся примеры алгоритмов преобразования Фурье с меньшей сложностью, чем у наилучших аналогов.

1 Введение

Известен ряд алгоритмов быстрого преобразования Фурье (БПФ) над полями вещественных и комплексных чисел, но перенос этих алгоритмов в конечные поля не всегда возможен. Кроме того, алгоритм БПФ, построенный специально для данного конечного поля, может оказаться лучше алгоритма, перенесенного из другого поля [1].

Предлагаемый метод состоит в разбиении исходного многочлена на сумму линеаризованных многочленов (1) и вычислении их значений в наборе базисных точек (2). Компоненты преобразования Фурье вычисляются как линейные комбинации этих значений с коэффициентами из простого поля (3).

Подход, основанный на представлении многочлена в виде суммы линеаризованных многочленов впервые был предложен в работе [2] и обобщен в [3]. Далее будут рассмотрены основные понятия и определения, введено циклотомическое разложение многочленов и представлен алгоритм БПФ, основанный на этом разложении. Алгоритм описывается для полей характеристики 2, но может быть обобщен и на случай произвольных конечных полей.

2 Основные понятия и определения

Определение 1. Преобразованием Фурье многочлена $f(x) = \sum\limits_{i=0}^{n-1} f_i x^i$ степени $\deg f(x) = n-1$, $n \mid (2^m - 1)$, в поле $GF(2^m)$ называется набор значений

$$F_j = f(\alpha^j) = \sum_{i=0}^{n-1} f_i \alpha^{ij}, \quad j \in [0, n-1],$$

где α – элемент порядка n в поле $GF(2^m)$.

Определение 2. Линеаризованными многочленами над полем $GF(2^m)$ называются многочлены вида

$$L(x) = \sum_i l_i x^{2^i}, \quad l_i \in GF(2^m).$$

Несложно показать, что для линеаризованных многочленов выполняется равенство $L(a+b) = L(a) + L(b)$. Следствием этого свойства является следующая теорема, приведенная здесь в модифицированном виде.

Теорема 1 ([4, 7]). *Пусть $x \in GF(2^m)$, и пусть элементы $\beta_0, \beta_1, \ldots, \beta_{m-1}$ образуют базис поля.*

$$\textit{Если} \quad x = \sum_{i=0}^{m-1} x_i \beta_i, \quad x_i \in GF(2), \qquad \textit{то} \quad L(x) = \sum_{i=0}^{m-1} x_i L(\beta_i).$$

Рассмотрим набор циклотомических классов по модулю n над $GF(2)$:

$$\{0\}, \{k_1, k_1 2, k_1 2^2, \ldots, k_1 2^{m_1 - 1}\}, \ldots, \{k_l, k_l 2, k_l 2^2, \ldots, k_l 2^{m_l - 1}\},$$

где $k_i \equiv k_i 2^{m_i} \bmod n$.

Многочлен $f(x) = \sum\limits_{i=0}^{n-1} f_i x^i, f_i \in GF(2^m)$, может быть разложен как

$$f(x) = \sum_{i=0}^{l} L_i(x^{k_i}), \quad L_i(y) = \sum_{j=0}^{m_i - 1} f_{k_i 2^j \bmod n} y^{2^j}. \tag{1}$$

Действительно, выражение (1) представляет собой способ группировки чисел $s \in \in [0, n-1]$ по циклотомическим классам: $s \equiv k_i 2^j \bmod n$. Очевидно, что такое разложение существует всегда. Заметим, что при $k_i = 0$ свободный член f_0 мы можем записать как значение многочлена $L_0(y) = f_0 y$ при $y = x^0$.

Выражение (1) будем называть циклотомическим разложением многочлена $f(x)$.

Пример 1. Многочлен $f(x) = \sum_{i=0}^{6} f_i x^i$, $f_i \in GF(2^3)$, представляется как

$$f(x) = L_0(x^0) + L_1(x) + L_2(x^3);$$
$$L_0(y) = f_0 y,$$
$$L_1(y) = f_1 y + f_2 y^2 + f_4 y^4,$$
$$L_2(y) = f_3 y + f_6 y^2 + f_5 y^4.$$

3 Быстрое вычисление преобразования Фурье

В соответствии с разложением (1) запишем $f(\alpha^j) = \sum_{i=0}^{l} L_i(\alpha^{j\,k_i})$. Как известно [5], элемент α^{k_i} является корнем соответствующего минимального многочлена степени m_i, и следовательно, лежит в подполе $GF(2^{m_i})$, $m_i \mid m$. Таким образом, все величины $(\alpha^{k_i})^j$ принадлежат полю $GF(2^{m_i})$ и могут быть разложены в каком-либо базисе $(\beta_{i,0}, \ldots, \beta_{i,m_i-1})$ этого поля: $\alpha^{j\,k_i} = \sum_{s=0}^{m_i-1} a_{ijs}\beta_{i,s}$, $a_{ijs} \in GF(2)$. Тогда значения каждого из линеаризованных многочленов могут быть вычислены в базисных точках соответствующего подполя по формуле

$$L_i(\beta_{i,s}) = \sum_{p=0}^{m_i-1} \beta_{i,s}^{2^p} f_{k_i 2^p}, \quad i \in [0, l], \quad s \in [0, m_i - 1]. \tag{2}$$

Базисы $(\beta_{i,0}, \ldots, \beta_{i,m_i-1})$ для каждого из линеаризованных многочленов $L_i(y)$ могут выбираться независимо.

В соответствии с теоремой 1 компоненты преобразования Фурье многочлена $f(x)$ являются линейными комбинациями этих значений:

$$F_j = f(\alpha^j) = \sum_{i=0}^{l} \sum_{s=0}^{m_i-1} a_{ijs} L_i(\beta_{i,s}) =$$
$$= \sum_{i=0}^{l} \sum_{s=0}^{m_i-1} a_{ijs} \left(\sum_{p=0}^{m_i-1} \beta_{i,s}^{2^p} f_{k_i 2^p} \right), \quad j \in [0, n-1]. \tag{3}$$

Последнее выражение может быть записано в матричной форме как $F = ALf$, где $F = (F_0, F_1, \ldots, F_{n-1})^T$; $f = (f_0, f_{k_1}, f_{k_1 2}, f_{k_1 2^2}, \ldots, f_{k_1 2^{m_1-1}}, \ldots, f_{k_l}, f_{k_l 2}, f_{k_l 2^2}, \ldots$ $\ldots, f_{k_l 2^{m_l-1}})^T$ есть перестановка вектора коэффициентов исходного многочлена $f(x)$, соответствующая разложению (1); A – матрица, составленная из элементов $a_{ijs} \in$ $\in GF(2)$; L – блочно-диагональная матрица, составленная из элементов $\beta_{i,s}^{2^p}$. Очевидно, что для линеаризованных многочленов одинаковой степени m_i, входящих в

разложение (1), можно выбрать одинаковые базисы $(\beta_{i,s})$ в подполях $GF(2^{m_i})$, вследствие чего матрица L будет содержать большое число одинаковых блоков.

Таким образом, задача БПФ разбивается на два этапа: умножение блочно-диагональной матрицы L на исходный вектор f и умножение двоичной матрицы A на полученный вектор $S = Lf$:

$$F = ALf. \tag{4}$$

Рассмотрим более подробно первый этап преобразования Фурье – задачу вычисления произведения $S = Lf$. Блочно-диагональная матрица

$$L = \begin{pmatrix} L_0 & 0 & \dots & 0 \\ 0 & L_1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & L_l \end{pmatrix}$$

состоит из блоков

$$L_i = \begin{pmatrix} \beta_{i,0} & \beta_{i,0}^2 & \dots & \beta_{i,0}^{2^{m_i-1}} \\ \beta_{i,1} & \beta_{i,1}^2 & \dots & \beta_{i,1}^{2^{m_i-1}} \\ \dots & \dots & \dots & \dots \\ \beta_{i,m_i-1} & \beta_{i,m_i-1}^2 & \dots & \beta_{i,m_i-1}^{2^{m_i-1}} \end{pmatrix}.$$

Пример использования стандартного базиса в качестве $(\beta_{i,0}, \dots, \beta_{i,m_i-1})$ рассмотрен в [6]. В случае нормального базиса $\left(\gamma_i, \gamma_i^2, \dots, \gamma_i^{2^{m_i-1}}\right)$ матрица L состоит из блоков вида

$$L_i = \begin{pmatrix} \gamma_i^{2^0} & \gamma_i^2 & \dots & \gamma_i^{2^{m_i-1}} \\ \gamma_i^2 & \gamma_i^4 & \dots & \gamma_i^{2^0} \\ \dots & \dots & \dots & \dots \\ \gamma_i^{2^{m_i-1}} & \gamma_i^{2^0} & \dots & \gamma_i^{2^{m_i-2}} \end{pmatrix}.$$

В силу блочно-диагональной структуры матрицы L вычисление произведения $S = Lf$ может быть представлено как $S = (b_0, b_1, \dots, b_l)^T = L\,(a_0, a_1, \dots, a_l)^T$, где $b_i = (b_{i,0}, b_{i,1}, \dots, b_{i,m_i-1})$ – подвектора искомого вектора S, $a_i = (a_{i,0}, a_{i,1}, \dots$ $\dots, a_{i,m_i-1})$ – подвектора исходного вектора f.

Представим вычисление $b_i^T = L_i a_i^T$ как циклическую свертку

$$b_i(x) = b_{i,0} + b_{i,m_i-1}x + \ldots + b_{i,1}x^{m_i-1} =$$
$$= (\gamma_i + \gamma_i^{2^{m_i-1}}x + \ldots + \gamma_i^2 x^{m_i-1})(a_{i,0} + a_{i,1}x + \ldots + a_{i,m_i-1}x^{m_i-1}) \bmod (x^{m_i} - 1).$$

Для ее вычисления могут быть применены известные алгоритмы [1, 7, 8]. При этом использование свойства нормального базиса $\gamma_i + \gamma_i^2 + \ldots + \gamma_i^{2^{m_i-1}} = 1$ позволяет заметно сократить число операций при вычислении циклической свертки. Отметим, что

вычисление значений линеаризованных многочленов с помощью циклической свертки было описано в монографии [7].

Описанный подход позволяет свести задачу умножения блочно-диагональной матрицы L на исходный вектор f над $GF(2^m)$ к задаче вычисления $l+1$ циклических сверток малой длины m_i. Существующие алгоритмы вычисления циклических сверток $b_i(x) = \gamma_i(x)a_i(x) \bmod (x^{m_i} - 1)$ могут быть записаны в матричном виде как

$$b_i = \begin{pmatrix} b_{i,0} \\ b_{i,1} \\ \dots \\ b_{i,m_i-1} \end{pmatrix} = Q_i \left(D_i \begin{pmatrix} \gamma_i \\ \gamma_i^{2^{m_i-1}} \\ \dots \\ \gamma_i^2 \end{pmatrix} \cdot (P_i a_i) \right),$$

где Q_i, D_i и P_i являются двоичными матрицами, а "$\cdot$" обозначает покомпонентное произведение векторов. Очевидно, что вектор $C_i = D_i \left(\gamma_i, \gamma_i^{2^{m_i-1}}, \dots, \gamma_i^2\right)^T$ может быть вычислен заранее. Таким образом, выражение (4) может быть переписано как

$$F = AQ\,(C \cdot (Pf)), \tag{5}$$

где Q – двоичная блочно-диагональная матрица объединенных последующих сложений для $l+1$ циклической свертки, C – объединенный вектор констант, P – двоичная блочно-диагональная матрица объединенных предварительных сложений.

Учитывая формулы (4) и (5), второй этап БПФ может рассматриваться как умножение двоичной матрицы AQ на вектор $C \cdot (Pf)$. Для вычисления произведения $(AQ)\,(C \cdot (Pf))$ могут быть использованы модифицированный алгоритм "четырех русских" (В.Л. Арлазаров, Е.А. Диниц, М.А. Кронрод, И.А. Фараджев) для умножения булевых матриц со сложностью $O(n^2/\log n)$ сложений над элементами поля $GF(2^m)$ [9] или эвристический алгоритм, сложность которого оценить не удалось. Однако для всех рассмотренных примеров сложность эвристического алгоритма меньше сложности модификации алгоритма четырех русских.

Приведенные преобразования имеют много общего с [10]. Основные отличия состоят в следующем:

1. Матрица L имеет регулярную структуру. Это позволяет свести задачу минимизации числа умножений к классической задаче вычисления циклической свертки.
2. Алгоритм содержит всего два умножения двоичных матриц на векторы, что может быть использовано для более глубокой оптимизации.
3. Используется более эффективный алгоритм оптимизации последовательности сложений.

Предложенный алгоритм БПФ эффективен при малых значениях длины преобразования (см. таблицу), хотя известны асимптотически более эффективные алгоритмы БПФ для конечных полей со сложностью $O(n \log^2 n)$ операций в основном поле [11, 12].

4 Пример

Пример 2. Продолжим рассмотрение БПФ длины 7 над полем $GF(2^3)$. Пусть α – корень примитивного многочлена $x^3 + x + 1$. В качестве базиса поля $GF(2^3)$ выберем нормальный базис $(\gamma, \gamma^2, \gamma^4)$, где $\gamma = \alpha^3$. Разложим многочлен $f(x)$ как в примере 1 и представим компоненты преобразования Фурье в виде сумм:

$$
\begin{gathered}
f(\alpha^0) = L_0(\alpha^0) + L_1(\alpha^0) + L_2(\alpha^0) = L_0(1) + L_1(\gamma) + L_1(\gamma^2) + L_1(\gamma^4) + \\
+ L_2(\gamma) + L_2(\gamma^2) + L_2(\gamma^4), \\
f(\alpha^1) = L_0(\alpha^0) + L_1(\alpha) + L_2(\alpha^3) = L_0(1) + L_1(\gamma^2) + L_1(\gamma^4) + L_2(\gamma), \\
f(\alpha^2) = L_0(\alpha^0) + L_1(\alpha^2) + L_2(\alpha^6) = L_0(1) + L_1(\gamma) + L_1(\gamma^4) + L_2(\gamma^2), \\
f(\alpha^3) = L_0(\alpha^0) + L_1(\alpha^3) + L_2(\alpha^2) = L_0(1) + L_1(\gamma) + L_2(\gamma) + L_2(\gamma^4), \\
f(\alpha^4) = L_0(\alpha^0) + L_1(\alpha^4) + L_2(\alpha^5) = L_0(1) + L_1(\gamma) + L_1(\gamma^2) + L_2(\gamma^4), \\
f(\alpha^5) = L_0(\alpha^0) + L_1(\alpha^5) + L_2(\alpha) = L_0(1) + L_1(\gamma^4) + L_2(\gamma^2) + L_2(\gamma^4), \\
f(\alpha^6) = L_0(\alpha^0) + L_1(\alpha^6) + L_2(\alpha^4) = L_0(1) + L_1(\gamma^2) + L_2(\gamma) + L_2(\gamma^2).
\end{gathered}
$$

Эта система может быть записана в матричной форме как

$$
F = \begin{pmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} L_0(1) \\ L_1(\gamma) \\ L_1(\gamma^2) \\ L_1(\gamma^4) \\ L_2(\gamma) \\ L_2(\gamma^2) \\ L_2(\gamma^4) \end{pmatrix} = AS.
$$

Тогда задачу БПФ можно переписать в виде

$$F = A \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \gamma^1 & \gamma^2 & \gamma^4 & 0 & 0 & 0 \\ 0 & \gamma^2 & \gamma^4 & \gamma^1 & 0 & 0 & 0 \\ 0 & \gamma^4 & \gamma^1 & \gamma^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \gamma^1 & \gamma^2 & \gamma^4 \\ 0 & 0 & 0 & 0 & \gamma^2 & \gamma^4 & \gamma^1 \\ 0 & 0 & 0 & 0 & \gamma^4 & \gamma^1 & \gamma^2 \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ f_4 \\ f_3 \\ f_6 \\ f_5 \end{pmatrix}.$$

Первый этап алгоритма БПФ состоит в вычислении двух циклических сверток

$$\begin{pmatrix} b_{i,0} \\ b_{i,1} \\ b_{i,2} \end{pmatrix} = \begin{pmatrix} \gamma^1 & \gamma^2 & \gamma^4 \\ \gamma^2 & \gamma^4 & \gamma^1 \\ \gamma^4 & \gamma^1 & \gamma^2 \end{pmatrix} \begin{pmatrix} a_{i,0} \\ a_{i,1} \\ a_{i,2} \end{pmatrix}, \quad i = 1, 2,$$

где

$$S = \begin{pmatrix} L_0(1) \\ L_1(\gamma) \\ L_1(\gamma^2) \\ L_1(\gamma^4) \\ L_2(\gamma) \\ L_2(\gamma^2) \\ L_2(\gamma^4) \end{pmatrix} = \begin{pmatrix} b_{0,0} \\ b_{1,0} \\ b_{1,1} \\ b_{1,2} \\ b_{2,0} \\ b_{2,1} \\ b_{2,2} \end{pmatrix}, \qquad f = \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ f_4 \\ f_3 \\ f_6 \\ f_5 \end{pmatrix} = \begin{pmatrix} a_{0,0} \\ a_{1,0} \\ a_{1,1} \\ a_{1,2} \\ a_{2,0} \\ a_{2,1} \\ a_{2,2} \end{pmatrix}.$$

Используя алгоритм вычисления трехточечной циклической свертки $b_i(x) = b_{i,0} + {} + b_{i,2}x + b_{i,1}x^2 = (\gamma + \gamma^4 x + \gamma^2 x^2)(a_{i,0} + a_{i,1}x + a_{i,2}x^2) \bmod (x^3 - 1)$, представленный в [1], получим

$$b_i = \begin{pmatrix} b_{i,0} \\ b_{i,1} \\ b_{i,2} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix} \left(\left[\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} \gamma \\ \gamma^4 \\ \gamma^2 \end{pmatrix} \right] \cdot \left[\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} a_{i,0} \\ a_{i,1} \\ a_{i,2} \end{pmatrix} \right] \right) =$$
$$= Q_i \left(C_i \cdot (P_i a_i) \right), \quad i = 1, 2.$$

С учетом $\gamma + \gamma^2 + \gamma^4 = 1$ видно, что алгоритм требует 3 умножения, 4 предварительных и 5 последующих сложений.

Теперь можно записать формулу (5) для рассматриваемого примера в матричной

форме:

$$F = \begin{pmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \end{pmatrix} = \left(\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} \right) \times$$

$$\times \left(\begin{pmatrix} 1 \\ 1 \\ \gamma^2+\gamma^4 \\ \gamma+\gamma^4 \\ \gamma+\gamma^2 \\ 1 \\ \gamma^2+\gamma^4 \\ \gamma+\gamma^4 \\ \gamma+\gamma^2 \end{pmatrix} \cdot \left[\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ f_4 \\ f_3 \\ f_6 \\ f_5 \end{pmatrix} \right] \right) = (AQ)(C \cdot (Pf)).$$

Второй этап БПФ состоит в умножении двоичной матрицы AQ на вектор $C \cdot (Pf)$:

$$F = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \end{pmatrix} (C \cdot (Pf)).$$

Этот этап может быть выполнен за 17 сложений.

Таким образом, БПФ длины 7 сводится к следующей последовательности действий:

– выполнение предварительных сложений $P \times f$:

$$\begin{array}{ll} V_1 = f_2 + f_4, & V_8 = f_6 + f_5, \\ V_2 = f_1 + f_2, & V_9 = f_3 + f_6, \\ V_3 = f_1 + f_4, & V_{10} = f_3 + f_5, \\ V_4 = f_1 + V_1, & V_{11} = f_3 + V_8; \end{array}$$

– выполнение умножений на константы $C \cdot (Pf)$:

$$\begin{array}{ll} V_5 = V_1\,\alpha, & V_{12} = V_8\,\alpha, \\ V_6 = V_2\,\alpha^2, & V_{13} = V_9\,\alpha^2, \\ V_7 = V_3\,\alpha^4, & V_{14} = V_{10}\,\alpha^4; \end{array}$$

– умножение матрицы AQ на вектор $C \cdot (Pf)$:

$$
\begin{array}{ll}
T_{10} = V_{12} + V_{14}, & F_2 = T_8 + T_{11}, \\
T_{11} = f_0 + V_{11}, & F_3 = T_7 + T_{14}, \\
T_{14} = f_0 + V_4, & T_{12} = F_2 + T_{10}, \\
T_{15} = V_5 + V_6, & T_{13} = F_3 + T_{15}, \\
T_{16} = V_6 + V_{13}, & F_4 = T_7 + T_{12}, \\
F_0 = V_4 + T_{11}, & F_5 = T_{10} + T_{13}, \\
T_9 = V_{12} + T_{16}, & F_6 = F_5 + T_7, \\
T_7 = V_7 + T_9, & F_1 = F_4 + T_8. \\
T_8 = V_5 + T_9, &
\end{array}
$$

Общая сложность алгоритма составляет $2 \times 3 = 6$ умножений и $2 \times 4 + 17 = 25$ сложений, что на одно сложение меньше, чем для алгоритма, представленного в [10].

5 Сравнение сложности алгоритмов БПФ

Вычисление $a + b$ (или $a \times b$) будем считать сложением (умножением) только тогда, когда оба слагаемых (сомножителя) принадлежат основному полю [13], т.е. операции в простом подполе не учитываются [1]. В таблице приведена сложность БПФ длины $n = 2^m - 1$ над полями $GF(2^m)$ в числе умножений N_{mul} и сложений N_{add}. Схема Горнера описана, например, в [7], а модификация алгоритма Герцеля для конечных полей рассмотрена в монографии [1]. Все алгоритмы, предложенные авторами в статье, доведены до программной реализации.

Предлагаемый метод был представлен на восьмой международной конференции “Алгебраическая и комбинаторная теория кодирования” (Царское Село, Россия), на семинаре кафедры информационных систем Санкт-Петербургского государственного университета аэрокосмического приборостроения и семинаре Института проблем передачи информации РАН (Москва). Авторы благодарят организаторов конференции и участников семинаров, а также рецензента статьи за сообщение о публикациях [11, 12]. Использованный при построении таблицы алгоритм циклической свертки длины 8 над $GF(2)$ был предложен Н. Чурковым. Кроме того, второй автор (С.В. Федоренко) благодарит фонд им. Александра фон Гумбольдта за многолетнюю поддержку его научных исследований.

Список литературы

[1] *Блейхут Р.* Теория и практика кодов, контролирующих ошибки. М.: Мир, 1986.

[2] *Truong T.-K., Jeng J.-H., Reed I.S.* Fast Algorithm – Computing the Roots of Error Locator Polynomials up to Degree 11 in Reed-Solomon Decoders // IEEE Transactions on Communications. 2001. V. 49. № 5. P. 779–783.

[3] *Fedorenko S.V., Trifonov P.V.* Finding Roots of Polynomials over Finite Fields // IEEE Transactions on Communications. 2002. V. 50. № 11. P. 1709–1711.

[4] *Берлекэмп Э.* Алгебраическая теория кодирования. М.: Мир, 1971.

[5] *Мак-Вильямс Ф.Дж., Слоэн Н.Дж.* Теория кодов, исправляющих ошибки. М.: Связь, 1979.

[6] *Fedorenko S., Trifonov P.* On Computing the Fast Fourier Transform over Finite Fields // Proc. Eighth Int. Workshop on Algebraic and Combinatorial Coding Theory. Tsarskoe Selo, Russia. September 2002. P. 108–111.

[7] *Габидулин Э.М., Афанасьев В.Б.* Кодирование в радиоэлектронике. М.: Радио и связь, 1986.

[8] *Афанасьев В.Б., Грушко И.И.* Алгоритмы БПФ для полей $GF(2^m)$ // Помехоустойчивое кодирование и надежность ЭВМ. М.: Наука, 1987. С. 33–55.

[9] *Ахо А., Хопкрофт Дж., Ульман Дж.* Построение и анализ вычислительных алгоритмов. М.: Мир, 1979.

[10] *Захарова Т.Г.* Вычисление преобразования Фурье в полях характеристики 2 // Пробл. передачи информ. 1992. Т. 28. № 2. С. 62–77.

[11] *Wang Y., Zhu X.* A Fast Algorithm for the Fouier Transform over Finite Fields and Its VLSI Implementation // IEEE J. on Selected Areas in Communications. 1988. V. 6. № 3. P. 572–577.

[12] *Afanasyev V.* On Complexity of FFT over Finite Field // Proc. Sixth Joint Swedish-Russian Int. Workshop on Information Theory. Molle, Sweden. August 1993. P. 315–319.

[13] *Блейхут Р.* Быстрые алгоритмы цифровой обработки сигналов. М.: Мир, 1989.

Трифонов Петр Владимирович
Федоренко Сергей Валентинович
Санкт-Петербургский государственный
политехнический университет
sfedorenko@ieee.org

Сложность некоторых алгоритмов БПФ

Параметр	Метод Горнера		Алгоритм Герцеля		Алгоритмы из [7] и [8]		Метод Захаровой [10]		Предлагаемый метод	
n	N_{mul}	N_{add}	N_{mul}	N_{add}	N_{mul}	N_{add}	N_{mul}	N_{add}	N_{mul}	N_{add}
7	36	42	12	42	9	35	6	26	6	25
15	196	210	38	210	20	70	16	100	16	77
31	900	930	120	930	108	645	60	388	54	315
63	3844	3906	282	3906	158	623	97	952	97	805
127	15876	16002	756	16002	594	5770	468	3737	216	2780
255	64516	64770	1718	64770	1225	4715	646	35503	586	7919
511	260100	260610	4044	260610	4374	–	–	–	1014	26643