УДК: 519.6

А.Н. Терещенко, В.К. Задирака

ОПТИМИЗАЦИЯ УМНОЖЕНИЯ БОЛЬШИХ N-РАЗРЯДНЫХ ЧИСЕЛ НА ОСНОВЕ N-РАЗРЯДНЫХ ДПФ

Рассматривается операция умножения больших чисел, от быстродействия которой зависит быстродействие ассиметричной криптографии. Приведено детальное описание алгоритма реализации операции умножения N-разрядных чисел на основе вычисления N-разрядных БПФ с использованием операций “распаковки” и “упаковки”. Описана процедура, позволяющая строить более простой алгоритм с использованием только формул “распаковки” или только формул “упаковки”.

Введение

Характерной особенностью решения многих задач аппроксимации функций, моделирования физических, химических (биохимических) процессов, аэродинамики, гидродинамики, защиты информации является использование вычислений над многоразрядными числами. Это обуславливает актуальность создания эффективных алгоритмов выполнения операций над многоразрядными числами для последующей программной реализации на универсальных ЭВМ и для специализированных аппаратных и программно-аппаратных комплексов. В данной статье рассматривается эффективная реализация операции умножения с использованием дискретного преобразования Фурье (ДПФ) [1–5], которое связано с быстрым вычислением циклической свертки двух дискретных сигналов. При умножении двух N-разрядных чисел получается $2N$-разрядный результат, из-за чего необходимо вычислять $2N$-разрядную свертку и $2N$-разрядные ДПФ множителей. В данной работе, которая является продолжением [6], описан эффективный метод вычисления операции умножения с использованием ДПФ, при котором разрядность исходного N-разрядного сигнала не изменяется, а для вычисления результата умножения достаточно знать $N+1$ первых значений $2N$-разрядной свертки. Для этого приводятся формулы перехода ДПФ от разрядности N к ДПФ разрядности $2N$ и наоборот. Метод [6] позволяет уменьшить число комплексных операций умножения приблизительно в 2 раза по сравнению со стандартным алгоритмом умножения на основе БПФ, при котором необходимо вычислять все $2N$ значений свертки. В работе приведена оптимизация метода [6], позволяющая упростить алгоритмическую (программную) реализацию операции умножения больших N-разрядных чисел.

Приведем два свойства ДПФ сигналов (см. [6]), которые позволяют упростить реализацию операции умножения.

Лемма 1 [6]. (Распаковка. Переход от меньшей разрядности N к большей $2N$). Четные и нечетные компоненты первых $N+1$ разрядов ДПФ $\hat{Z}_{2N}$ действительного сигнала $Z_{2N}(r)$, $r=\overline{0,2N-1}$, могут быть вычислены на основе ДПФ $\hat{X}_N$ сигнала $X_N(r)=Z_{2N}(2r)+iZ_{2N}(2r+1)$, $r=\overline{0,N-1}$, используя следующие соотношения:

$$\hat{Z}_{2N}(0)=\operatorname{Re}\hat{X}_N(0)+\operatorname{Im}\hat{X}_N(0),$$

$$Z_{2N}(N)=\operatorname{Re}X_N(0)-\operatorname{Im}X_N(0),$$

$$\hat{Z}_{2N}(N/2)=\hat{X}_N^*(N/2).$$

$$\hat{Z}_{2N}(r)=A_{N/2}(r)+S_{N/2}(r),$$

$$\hat{Z}_{2N}(N-r)=A_{N/2}^*(r)-S_{N/2}^*(r),$$

$$A_{N/2}(r)=\frac{1}{2}\left(\hat{X}_N(r)+\hat{X}_N^*(N-r)\right),$$

$$S_{N/2}(r)=\frac{W_{2N}^r}{2i}\left(\hat{X}_N(r)-\hat{X}_N^*(N-r)\right),$$

$$W_{2N}^r=e^{-\frac{2\pi i}{2N}r}=e^{-\frac{\pi i}{N}r},\ r=\overline{1,N/2-1}.$$

Лемма 2 [6]**.** (Упаковка. Переход от большей разрядности $2N$ к меньшей N). В условиях Леммы 1 справедливы следующие соотношения:

$$\hat{X}_N(0)=\frac{1}{2}\Big(\big(\operatorname{Re}\hat{Z}_{2N}(0)+\operatorname{Re}\hat{Z}_{2N}(N)\big)+$$
$$+i\big(\operatorname{Re}\hat{Z}_{2N}(0)-\operatorname{Re}\hat{Z}_{2N}(N)\big)\Big),$$
$$\hat{X}_N(N/2)=\hat{Z}^*_{2N}(N/2).$$
$$\hat{X}_N(r)=A_{N/2}(r)-S_{N/2}(r),$$
$$\hat{X}_N(N-r)=A^*_{N/2}(r)+S^*_{N/2}(r),$$
$$A_{N/2}(r)=\frac{1}{2}\Big(\hat{Z}_{2N}(r)+\hat{Z}^*_{2N}(N-r)\Big),$$
$$S_{N/2}(r)=\frac{W^{-r}_{2N}}{2i}\Big(Z_{2N}(r)-Z^*_{2N}(N-r)\Big),$$
$$W^r_{2N}=e^{-\frac{\pi i}{N}r},\ r=\overline{1,N/2-1}. \qquad (1)$$

Алгоритм 1. Реализация операции умножения двух N-разрядных сигналов на основе N-разрядных ДПФ с использованием формул распаковки и упаковки. Приведем его пошаговое описание.

Вход: $U_N(r)$, $V_N(r)$, $r=\overline{0,N-1}$ – N-разрядные множители.

Выход: R_{2N} – $2N$-разрядный результат умножения.

Шаг 1. Предвычисление векторов $B_N(r)$, $C_N(r)$, $r=\overline{0,N-1}$.

$$B_N(0)\leftarrow 0,\ C_N(0)\leftarrow 1,\ r\leftarrow 0,$$
$$n\leftarrow \log_2 N.$$
$$r\leftarrow r+1,\ C_N(r)\leftarrow\cos\left(-\frac{v}{2}\cdot\frac{2\pi}{N}\right),$$
$$B_N(r)\leftarrow v,\ v\leftarrow B_N(k)+d,\ k=\overline{0,j-1},$$
$$j\leftarrow 2^p,\ d\leftarrow 2^{n-p-1},\ p=\overline{0,n-1}.$$

Шаг 2. Инициализация сигналов, $\hat{Y}_N(r)$ из $U_N(r)$, $V_N(r)$, $r=\overline{0,N-1}$.

$$X_N(r)\leftarrow U_N(2r)+iU_N(2r+1),$$
$$Y_N(r)\leftarrow V_N(2r)+iV_N(2r+1),\ r=\overline{0,N/2-1}.$$
$$X_N(r)\leftarrow Y_N(r)\leftarrow 0,$$
$$r=\overline{N/2,N-1}.$$

Шаг 3. Вычисление БПФ сигналов $\hat{X}_N(r)$, $\hat{Y}_N(r)$, $r=\overline{0,N-1}$.

$$\hat{X}_N(i1)\leftarrow\hat{X}_N(i1)+X,\ \hat{X}_N(i2)\leftarrow\hat{X}_N(i1)-X,$$
$$\hat{Y}_N(i1)\leftarrow\hat{Y}_N(i1)+Y,\ \hat{Y}_N(i2)\leftarrow\hat{Y}_N(i1)-Y,$$
$$X\leftarrow X\cdot W,\ X\leftarrow\hat{X}_N(i2),\ Y\leftarrow Y\cdot W,$$
$$Y\leftarrow\hat{Y}_N(i2),$$

$$\begin{cases} W\leftarrow 1, & \text{при}\quad k=0\\ W\leftarrow -i, & \text{при}\quad k=1\\ W\leftarrow S-iS, & \text{при}\quad k=2\\ W\leftarrow -S-iS, & \text{при}\quad k=3\\ W\leftarrow\begin{cases}\langle k\rangle_2=0, & C_N(k)+iC_N(k+1)\\ \langle k\rangle_2=1, & C_N(k)-iC_N(k-1)\end{cases}, & \text{при}\quad k>3\end{cases},$$

$$S\leftarrow\frac{\sqrt{2}}{2},\ i1\leftarrow s1+r,\ i2\leftarrow s2+r,$$
$$s2\leftarrow s1+d,\ s1\leftarrow k\cdot(2\cdot d),\ r=\overline{0,d-1},$$
$$k=\overline{0,j-1},\ d\leftarrow 2^{n-1-p},\ j\leftarrow 2^p,$$
$$p=\overline{0,n-1}.$$

Шаг 4. Битовая инверсия сигналов $\hat{X}_N(r)$, $\hat{Y}_N(r)$, $r=\overline{0,N-1}$.

$$\begin{cases} r<j, & \left\{\begin{array}{c} X\leftarrow\hat{X}_N(j),Y\leftarrow\hat{Y}_N(j);\\ \hat{X}_N(j)\leftarrow\hat{X}_N(r),\hat{Y}_N(j)\leftarrow\hat{Y}_N(r);\\ \hat{X}_N(r)\leftarrow X,\hat{Y}_N(r)\leftarrow Y.\end{array}\right\}\\ r\ge j, & \hat{X}_N(r)\leftarrow\hat{X}_N(r),\hat{Y}_N(r)\leftarrow\hat{Y}_N(r)\end{cases},$$
$$j\leftarrow B_N(r),\ r=\overline{0,N-1}.$$

Шаг 5. Распаковка сигналов $\hat{X}_N(r)$, $\hat{Y}_N(r)$, $r=\overline{0,N-1}$.

$$\hat{X}_N(0)\leftarrow\operatorname{Re}X+\operatorname{Im}X,$$
$$\hat{Y}_N(0)\leftarrow\operatorname{Re}Y+\operatorname{Im}Y,$$
$$\hat{X}_N(N)\leftarrow\operatorname{Re}X-\operatorname{Im}X,\ X\leftarrow\hat{X}_N(0).$$
$$\hat{Y}_N(N)\leftarrow\operatorname{Re}Y-\operatorname{Im}Y,\ Y\leftarrow\hat{Y}_N(0).$$
$$\hat{X}_N(N/2)\leftarrow\hat{X}^*_N(N/2),$$
$$\hat{Y}_N(N/2)\leftarrow\hat{Y}^*_N(N/2).$$
$$\hat{X}_N(r)\leftarrow AX+SX,\ \hat{Y}_N(r)\leftarrow AY+SY,$$
$$\hat{X}_N(N-r)\leftarrow AX^*-SX^*,\ SX\leftarrow W\cdot SX,$$
$$\hat{Y}_N(N-r)\leftarrow AY^*-SY^*,\ SY\leftarrow W\cdot SY,$$

$$AX \leftarrow \frac{1}{2}\left(\hat{X}_N(r) + \hat{X}_N^*(N-r)\right),$$

$$SX \leftarrow \frac{1}{2i}\left(\hat{X}_N(r) - \hat{X}_N^*(N-r)\right),$$

$$AY \leftarrow \frac{1}{2}(\hat{Y}_N(r) + \hat{Y}_N^*(N-r)),$$

$$SY \leftarrow \frac{1}{2i}(\hat{Y}_N - \hat{Y}_N^*(N-r)),$$

$$W \leftarrow \begin{cases} \langle k \rangle_2 = 0, & C_N(k) + iC_N(k+1) \\ \langle k \rangle_2 = 1, & C_N(k) - iC_N(k-1) \end{cases},$$

$$k \leftarrow B_N(r),\ r = \overline{0,\ N/2-1}.$$

Шаг 6. Комплексное умножение сигналов $\hat{X}_{N+1}$, $\hat{Y}_{N+1}$, $r = \overline{0,\ N}$.

$$\hat{Z}_{N+1}(r) \leftarrow \hat{X}_{N+1}(r) \cdot \hat{Y}_{N+1}(r),\ r = \overline{0,\ N}.$$

Шаг 7. Упаковка сигнала $\hat{Z}_{N+1}(r)$, $r = \overline{0,\ N}$.

$$\hat{Z}_N(0) \leftarrow \frac{1}{2}\Big(\big(\operatorname{Re}\hat{Z}_N(0) + \operatorname{Im}\hat{Z}_N(N)\big) + i\big(\operatorname{Re}\hat{Z}_N(0) - \operatorname{Im}\hat{Z}_N(N)\big)\Big),$$

$$\hat{Z}_N(N/2) \leftarrow \hat{Z}_N^*(N/2).$$

$$\hat{Z}_N(r) \leftarrow A - S,\ \hat{Z}_N(N-r) \leftarrow A^* + S^*,$$

$$S \leftarrow \left(W^*\right) \cdot S,\ A \leftarrow \frac{1}{2}\left(\hat{Z}_N(r) + \hat{Z}_N^*(N-r)\right),$$

$$S \leftarrow \frac{1}{2i}\left(\hat{Z}_N(r) - \hat{Z}_N^*(N-r)\right),$$

$$W \leftarrow \begin{cases} \langle k \rangle_2 = 0, & C_N(k) + iC_N(k+1) \\ \langle k \rangle_2 = 1, & C_N(k) - iC_N(k-1) \end{cases},$$

$$k \leftarrow B_N(r),\ r = \overline{0,\ N/2-1}.$$

Шаг 8. Комплексное сопряжение сигнала $\hat{Z}_N(r)$, $r = \overline{0,\ N-1}$.

$$\hat{Z}_N(r) \leftarrow \hat{Z}_N^*(r),\ r = \overline{0,\ N-1}.$$

Шаг 9. Вычисление БПФ сигнала $\hat{Z}_N(r)$, $r = \overline{0,\ N-1}$.

$$Z_N(i1) \leftarrow Z_N(i1) + Z,$$

$$Z_N(i2) \leftarrow Z_N(i1) - Z,\ Z \leftarrow Z \cdot W,$$

$$Z \leftarrow Z_N(i2),$$

$$\begin{cases} W \leftarrow 1, & \text{при} \quad k = 0 \\ W \leftarrow -i, & \text{при} \quad k = 1 \\ W \leftarrow S - iS, & \text{при} \quad k = 2 \\ W \leftarrow -S - iS, & \text{при} \quad k = 3 \\ W \leftarrow \begin{cases} \langle k \rangle_2 = 0, & C_N(k) + iC_N(k+1) \\ \langle k \rangle_2 = 1, & C_N(k) - iC_N(k-1) \end{cases}, \text{при} & k > 3 \end{cases},$$

$$S \leftarrow \sqrt{2}/2,\ i1 \leftarrow s1 + r,\ i2 \leftarrow s2 + r,$$

$$s2 \leftarrow s1 + d,\ s1 \leftarrow k \cdot (2 \cdot d),\ r = \overline{0, d-1},$$

$$k = \overline{0,\ j-1},\ d \leftarrow 2^{n-1-p},\ j \leftarrow 2^p,\ p = \overline{0, n-1}.$$

Шаг 10. Битовая инверсия сигналов Z_N, $r = \overline{0,\ N-1}$.

$$\begin{cases} r < j, & \begin{cases} Z_N(r) \leftarrow Z, \\ Z_N(j) \leftarrow Z_N(r), \\ Z \leftarrow Z_N(j) \end{cases} \\ r \geq j, & Z_N(r) \leftarrow Z_N(r) \end{cases},\ j \leftarrow B_N(r),$$

$r = \overline{0,\ N-1}$.

Шаг 11. Комплексное сопряжение сигнала $Z_N(r)$, $r = \overline{0,\ N-1}$.

$$Z_N(r) \leftarrow Z_N^*(r),\ r = \overline{0,\ N-1}.$$

Шаг 12. Результат умножения из сигнала $\hat{Z}_N$, $r = \overline{0,\ N-1}$.

$$R_{2N}(2r) \leftarrow \operatorname{Re} Z_N(r),$$

$$R_{2N}(2r+1) \leftarrow \operatorname{Im} Z_N(r),\ r = \overline{0,\ N-1}.$$

Примечание. На 5-м шаге (Распаковка) длина сигналов $\hat{X}_N$, $\hat{X}_N$ увеличивается на один разряд, а на 7-м шаге (Упаковка) длина сигнала $\hat{Z}_{N+1}$ уменьшается на один разряд.

Алгоритм 2. Реализация операции умножения двух N-разрядных сигналов на основе N-разрядных ДПФ с использованием формул распаковки и упаковки (в виде подалгоритмов).

Вход: $U_N(r)$, $V_N(r)$, $r = \overline{0,\ N-1}$ – N-разрядные множители,

N – число разрядов множителей.

Выход: R_{2N} – $2N$-разрядный результат умножения.

Шаг 1. $(C_N, B_N) \leftarrow$ Предвычисление (N)[1] (см. Алгоритм 3).

Шаг 2. $(X_N, Y_N) \leftarrow$ Инициализация (U_N, V_N, N) (см. Алгоритм 4).

Шаг 3. $\hat{X}_N \leftarrow$ БПФ (X_N, C_N, N) (см. Алгоритм 5).

Шаг 4. $\hat{Y}_N \leftarrow$ БПФ (Y_N, C_N, N) (см. Алгоритм 5).

Шаг 5. $\hat{X}_N \leftarrow$ Битовая инверсия $(\hat{X}_N, B_N, N)$ (см. Алгоритм 6).

Шаг 6. $\hat{Y}_N \leftarrow$ Битовая инверсия$(\hat{Y}_N, B_N, N)$ (см. Алгоритм 6).

Шаг 7. $\hat{X}_{N+1} \leftarrow$ Распаковка $(\hat{X}_N, C_N, B_N, N)$ (см. Алгоритм 7).

Шаг 8. $\hat{Y}_{N+1} \leftarrow$ Распаковка$(\hat{Y}_N, C_N, B_N, N)$ (см. Алгоритм 7).

Шаг 9. $\hat{Z}_{N+1} \leftarrow$ Умножение$(\hat{X}_{N+1}, \hat{Y}_{N+1}, N)$ (см. Алгоритм 8).

Шаг 10. $\hat{Z}_N \leftarrow$ Упаковка$(\hat{Z}_{N+1}, C_N, B_N, N)$ (см. Алгоритм 9).

Шаг 11. $\hat{Z}_N \leftarrow$ Комплексное сопряжение$(\hat{Z}_N, N)$ (см. Алгоритм 10).

Шаг 12. $Z_N \leftarrow$ БПФ$(\hat{Z}_N, C_N, N)$ (см. Алгоритм 5).

Шаг 13. $Z_N \leftarrow$ Битовая инверсия(Z_N, B_N, N) (см. Алгоритм 6).

Шаг 14. $Z_N \leftarrow$ Комплексное сопряжение(Z_N, N) (см. Алгоритм 10).

Шаг 15. $R_{2N} \leftarrow$ Результат умножения(Z_N, N) (см. Алгоритм 11).

Примечание. См. табл. 1–11. Результат вычисления Алгоритмов при $N = 16$.

Алгоритм 3. Предвычисление.

Вход: N – число разрядов комплексного сигнала.

Выход: $C_N(r)$, $r = \overline{0, N-1}$ – вектор предвычисленных косинусов,

$B_N(r)$, $r = \overline{0, N-1}$ – вектор номеров строк для битовой инверсии.

Шаг 1. $B_N(0) \leftarrow 0$; $C_N(0) \leftarrow 1$ (или $C_N(0) \leftarrow \cos(B_N(0))$); $r \leftarrow 0$, $n \leftarrow \log_2 N$.

Шаг 2. $j \leftarrow 1$, $d \leftarrow N/2$.

Шаг 3. Для p от 0 до $n-1$.

Шаг 4. Для k от 0 до $j-1$.

Шаг 5. $v \leftarrow B_N(k) + d$.

Шаг 6.

$$B_N(r) \leftarrow v,\ C_N(r) \leftarrow \cos\left(-\frac{v}{2} \cdot \frac{2\pi}{N}\right).$$

Шаг 7. $r \leftarrow r+1$.

Шаг 8. Конец цикла по k.

Шаг 9. $j \leftarrow 2 \cdot j$, $d \leftarrow d/2$.

Шаг 10. Конец цикла по p.

Алгоритм 4. Инициализация.

Вход: $U_N(r)$, $V_N(r)$, $r = \overline{0, N-1}$ – действительные сигналы,

N – число разрядов в действительных и комплексных сигналах.

Выход: $X_N(r)$, $Y_N(r)$, $r = \overline{0, N-1}$ – комплексные сигналы.

Шаг 1. Для r от 0 до $N/2-1$.

Шаг 2.

$$X_N(r) \leftarrow U_N(2r) + iU_N(2r+1).$$

Шаг 3. $Y_N(r) \leftarrow V_N(2r) + iV_N(2r+1)$.

Шаг 4. Конец цикла по r.

Шаг 5. Для r от $N/2$ до $N-1$.

Шаг 6. $x_r \leftarrow 0$, $y_r \leftarrow 0$.

Шаг 7. Конец цикла по r.

Алгоритм 5. БПФ (Быстрое преобразование Фурье) оптимизированный.

Вход: $\hat{X}_N(r)$, $r = \overline{0, N-1}$ – комплексный сигнал,

[1] Алгоритмы 3-11 описаны далее

$C_N(r)$, $r=\overline{0,N-1}$ – вектор предвычисленных косинусов (см. Алгоритм 3. Предвычисления),

N – число разрядов комплексного сигнала.

Выход: $\hat{X}_N(r)$, $r=\overline{0,N-1}$ – ДПФ входного комплексного сигнала $\hat{X}_N(r)$, $r=\overline{0,N-1}$.

Шаг 1. $S \leftarrow \sqrt{2}/2$, $j \leftarrow 1$, $d \leftarrow N/2$.

Шаг 2. Для p от 0 до $n-1$.

Шаг 3. Для k от 0 до $j-1$.

Шаг 4. $s1 \leftarrow k\cdot(2\cdot d)$, $s2 \leftarrow s1+d$.

Шаг 5. Для r от 0 до $d-1$.

Шаг 6. $i1 \leftarrow s1+r$, $i2 \leftarrow s2+r$.

Шаг 7. $X \leftarrow \hat{X}_N(i2)$;

$$\begin{cases} X \leftarrow X, & \text{при} \quad k=0 \\ X \leftarrow X\cdot(-i), & \text{при} \quad k=1 \\ X \leftarrow X\cdot(S-iS), & \text{при} \quad k=2 \\ X \leftarrow X\cdot(-S-iS), & \text{при} \quad k=3 \\ X \leftarrow X\cdot\begin{cases} \langle k\rangle_2=0, & C_N(k)+iC_N(k+1) \\ \langle k\rangle_2=1, & C_N(k)-iC_N(k-1) \end{cases}, & \text{при} \quad k>3 \end{cases}$$

Шаг 8. $\hat{X}_N(i2) \leftarrow \hat{X}_N(i1)-X$,

$$\hat{X}_N(i1) \leftarrow \hat{X}_N(i1)+X.$$

Шаг 9. Конец цикла по r.

Шаг 10. Конец цикла по k.

Шаг 11. Конце цикла по p.

Алгоритм 6. Битовая инверсия.

Вход: $\hat{X}_N(r)$, $r=\overline{0,N-1}$ – комплексный сигнал,

$B_N(r)$, $r=\overline{0,N-1}$ – вектор номеров строк для битовой инверсии (см. Алгоритм 3),

N – число разрядов комплексного сигнала.

Выход: $\hat{X}_N$ – результат битовой инверсии входного комплексного сигнала $\hat{X}_N$.

Шаг 1. Для r от 0 до $N-1$.

Шаг 2. $j \leftarrow B_N(r)$.

Шаг 3. Если $r<j$, то $\hat{X}_N(r) \leftarrow X$,

$\hat{X}_N(j) \leftarrow \hat{X}_N(r)$, $X \leftarrow \hat{X}_N(j)$.

Шаг 4. Конец цикла по r.

Алгоритм 7. Распаковка.

Вход: $\hat{X}_N(r)$, $r=\overline{0,N}$ – комплексный сигнал,

$C_N(r)$, $r=\overline{0,N}$ – вектор предвычисленных косинусов,

$B_N(r)$, $r=\overline{0,N}$ – вектор номеров строк для битовой инверсии (см. Алгоритм 3. Предвычисления),

N – число разрядов комплексного сигнала.

Выход: $\hat{X}_{N+1}(r)$, $r=\overline{0,N}$ – $N+1$ первых разрядов ДПФ действительного сигнала $U_N(r)$, $r=\overline{0,N}$, добавленного N старшими нулями.

Шаг 1. $\hat{X}_N(0) \leftarrow \operatorname{Re}X+\operatorname{Im}X$,

$$\hat{X}_N(N) \leftarrow \operatorname{Re}X-\operatorname{Im}X,$$

$$X \leftarrow \hat{X}_N(0).$$

Шаг 2. Для r от 1 до $\frac{N}{2}-1$.

Шаг 3. $k \leftarrow B_N(r)$;

$$W \leftarrow \begin{cases} \langle k\rangle_2=0, & C_N(k)+iC_N(k+1) \\ \langle k\rangle_2=1, & C_N(k)-iC_N(k-1) \end{cases}.$$

Шаг 4. $A \leftarrow \frac{1}{2}\left(\hat{X}_N(r)+\hat{X}_N^*(N-r)\right)$,

$$S \leftarrow \frac{1}{2}\left(\hat{X}_N(r)-\hat{X}_N^*(N-r)\right).$$

Шаг 5. $S \leftarrow (-i)\cdot W\cdot S$.

Шаг 6. $\hat{X}_N(r) \leftarrow A+S$,

$$\hat{X}_N(N-r) \leftarrow A^*-S^*.$$

Шаг 7. Конец цикла по r.

Алгоритм 8. Умножение.

Вход: $\hat{X}_{N+1}(r)$, $\hat{Y}_{N+1}(r)$, $r=\overline{0,N}$ – $N+1$ первых разрядов ДПФ действительных сигналов $U_N(r)$ и $V_N(r)$, $r=\overline{0,N}$, добавленных N старшими нулями;

N – число разрядов комплексного сигнала.

Выход: $\hat{Z}_{N+1}(r)$, $r=\overline{0,N}$ – результат умножения сигналов $\hat{X}_{N+1}(r)$, $\hat{Y}_{N+1}(r)$, $r=\overline{0,N}$.

Шаг 1.

$$\hat{Z}_{N+1}(0)\leftarrow \operatorname{Re}\hat{X}_{N+1}(0)\cdot\operatorname{Re}\hat{Y}_{N+1}(0),$$

$$\hat{Z}_{N+1}(N)\leftarrow \operatorname{Re}\hat{X}_{N+1}(N)\cdot\operatorname{Re}\hat{Y}_{N+1}(N).$$

Шаг 2. Для r от 1 до $N-1$.

Шаг 3. $\hat{Z}_{N+1}(r)\leftarrow \hat{X}_{N+1}(r)\cdot\hat{Y}_{N+1}(r)$.

Шаг 4. Конец цикла по r.

Алгоритм 9. Упаковка.

Вход: $\hat{Z}_{N+1}(r)$, $r=\overline{0,N}$ – $N+1$ первых разрядов умножения ДПФ действительных сигналов $U_N(r)$ и $V_N(r)$, $r=\overline{0,N-1}$, добавленных N старшими нулями.

$C_N(r)$, $r=\overline{0,N-1}$ – вектор предвычисленных косинусов,

$B_N(r)$, $r=\overline{0,N-1}$ – вектор номеров строк для битовой инверсии (см. Алгоритм 3. Предвычисления),

N – число разрядов комплексного сигнала.

Выход: $\hat{Z}_N(r)$, $r=\overline{0,N-1}$ – обратное ДПФ сигнала $Z_N(r)$, $r=\overline{0,N-1}$.

Шаг 1.

$$\hat{Z}_N(0)\leftarrow\frac{1}{2}\Big(\big(\operatorname{Re}\hat{Z}_N(0)+\operatorname{Re}\hat{Z}_N(N)\big)+ +i\big(\operatorname{Re}\hat{Z}_N(0)-\operatorname{Re}\hat{Z}_N(N)\big)\Big).$$

Шаг 2. Для r от 1 до $\frac{N}{2}-1$.

Шаг 3. $k\leftarrow B_N(r)$;

$$W\leftarrow\begin{cases}\langle k\rangle_2=0, & C_N(k)+iC_N(k+1)\\ \langle k\rangle_2=1, & C_N(k)-iC_N(k-1)\end{cases}.$$

Шаг 4. $A\leftarrow\frac{1}{2}\big(\hat{Z}_N(r)+\hat{Z}_N^*(N-r)\big)$,

$$S\leftarrow\frac{1}{2}\big(\hat{Z}_N(r)-\hat{Z}_N^*(N-r)\big).$$

Шаг 5. $S\leftarrow(-i)\cdot\big(W^*\big)\cdot S$.

Шаг 6. $\hat{Z}_N(r)\leftarrow A-S$,

$$\hat{Z}_N(N-r)\leftarrow A^*+S^*.$$

Шаг 7. Конец цикла по r.

Алгоритм 10. Комплексное сопряжение.

Вход: $\hat{X}_N(r)$, $r=\overline{0,N-1}$ – комплексный сигнал,

N – число разрядов комплексного сигнала.

Выход: $\hat{X}_N(r)$, $r=\overline{0,N-1}$ – результат комплексного сопряжения входного сигнала $\hat{X}_N(r)$, $r=\overline{0,N-1}$.

Шаг 1. Для r от 0 до $N-1$.

Шаг 2. $\hat{Z}_N(r)\leftarrow\hat{Z}_N^*(r)$.

Шаг 3. Конец цикла по r.

Алгоритм 11. Результат умножения.

Вход: $Z_N(r)$, $r=\overline{0,N-1}$ – комплексный сигнал,

N – число разрядов комплексного сигнала.

Выход: $R_{2N}(r)$, $r=\overline{0,2N-1}$ – действительный сигнал, результат умножения $U_N(r)$ и $V_N(r)$, $r=\overline{0,N-1}$.

Шаг 1. Для r от 0 до $N-1$.

Шаг 2. $R_{2N}(2r)\leftarrow\operatorname{Re}Z_N(r)/N$,

$$R_{2N}(2r+1)\leftarrow\operatorname{Im}Z_N(r)/N.$$

Шаг 3. Конец цикла по r.

Далее приводятся табл. 1 – 12 промежуточных результатов выполнения каждого шага Алгоритма 2 на примере операции умножения двух 16-разрядных чисел, все разряды которых одинаковые и равны 1, что соответствует операции возведения в квадрат 16-разрядного числа.

Таблица 1. Результат вычисления Алгоритма 3. Предвычисления ($N = 16$)

r	$C_N(r)$	$B_N(r)$
0	1	0
1	0	8
2	0.7071067812	4
3	–0.7071067812	12
4	0.9238795325	2
5	–0.3826834324	10
6	0.3826834324	6
7	–0.9238795325	14
8	0.9807852804	1
9	–0.195090322	9
10	0.555570233	5
11	–0.8314696123	13
12	0.8314696123	3
13	–0.555570233	11
14	0.195090322	7
15	–0.9807852804	15

Таблица 2. Результат вычисления Алгоритма 4. Инициализация ($N = 16$)

r	$U_N(r), V_N(r)$	$\operatorname{Re}\hat{X}_N(r), \operatorname{Re}\hat{Y}_N(r),$ $\operatorname{Im}\hat{X}_N(r), \operatorname{Im}\hat{Y}_N(r)$
0	1	1
1	1	1
2	1	1
3	1	1
4	1	1
5	1	1
6	1	1
7	1	1
8	1	0
9	1	0
10	1	0
11	1	0
12	1	0
13	1	0
14	1	0
15	1	0

Таблица 3. Результат итераций ($p = 0,1,2$) Алгоритма 5. БПФ сигналов $\hat{X}_N(r)$, $\hat{Y}_N(r)$

r	Вход		$p=0$		$p=1$		$p=2$		$p=3$	
	Re	Im	Re	Im	Re	Im	Re	Im	Re	Im
0	1	1	1	1	2	2	4	4	8	8
1	1	1	1	1	2	2	4	4	0	0
2	1	1	1	1	2	2	0	0	0	0
3	1	1	1	1	2	2	0	0	0	0
4	1	1	1	1	0	0	0	0	0	0
5	1	1	1	1	0	0	0	0	0	0
6	1	1	1	1	0	0	0	0	0	0
7	1	1	1	1	0	0	0	0	0	0
8	0	0	1	1	2	0	3.414213562	–1.414213562	6.027339492	–4.027339492
9	0	0	1	1	2	0	3.414213562	–1.414213562	0.8010876326	1.198912367
10	0	0	1	1	2	0	0.5857864376	1.414213562	1.668178638	0.3318213621
11	0	0	1	1	2	0	0.5857864376	1.414213562	–0.4966057627	2.496605763
12	0	0	1	1	0	2	1.414213562	0.5857864376	2.496605763	–0.4966057627
13	0	0	1	1	0	2	1.414213562	0.5857864376	0.3318213621	1.668178638
14	0	0	1	1	0	2	–1.414213562	3.414213562	1.198912367	0.8010876326
15	0	0	1	1	0	2	–1.414213562	3.414213562	–4.027339492	6.027339492

Таблица 4. Результат Алгоритма 6. Битовая инверсия сигналов $\hat{X}_N(r)$, $\hat{Y}_N(r)$

r	Вход $\hat{X}_N(r)$		Выход $\mathrm{Im}\,\hat{X}_N(r)$	
	Re	Im	Re	Im
0	8	8	8	8
1	0	0	6.027339492	–4.027339492
2	0	0	0	0
3	0	0	2.496605763	–0.4966057627
4	0	0	0	0
5	0	0	1.668178638	0.3318213621
6	0	0	0	0
7	0	0	1.198912367	0.8010876326
8	6.027339492	–4.027339492	0	0
9	0.8010876326	1.198912367	0.8010876326	1.198912367
10	1.668178638	0.3318213621	0	0
11	–0.4966057627	2.496605763	0.3318213621	1.668178638
12	2.496605763	–0.4966057627	0	0
13	0.3318213621	1.668178638	–0.4966057627	2.496605763
14	1.198912367	0.8010876326	0	0
15	–4.027339492	6.027339492	–4.027339492	6.027339492

Таблица 5. Результат Алгоритма 7. Распаковка $\hat{X}_N(r)$, $\hat{Y}_N(r)$

r	Вход $\hat{X}_N(r)$		Выход $\hat{X}_{N+1}(r)$	
	Re	Im	Re	Im
0	8	8	16	0
1	6.027339492	–4.027339492	1	–10.15317039
2	0	0	0	0
3	2.496605763	–0.4966057627	1	–3.296558209
4	0	0	0	0
5	1.668178638	0.3318213621	1	–1.870868412
6	0	0	0	0
7	1.198912367	0.8010876326	1	–1.218503526
8	0	0	0	0
9	0.8010876326	1.198912367	1	–0.8206787908
10	0	0	0	0
11	0.3318213621	1.668178638	1	–0.534511136
12	0	0	0	0
13	–0.4966057627	2.496605763	1	–0.3033466836
14	0	0	0	0
15	–4.027339492	6.027339492	1	–0.09849140336
16			0	0

Таблица 6. Результат Алгоритма 8. Умножение $\hat{X}_{N+1}(r) \cdot \hat{Y}_{N+1}(r)$

r	Вход $\hat{X}_{N+1}(r)$, $\hat{Y}_{N+1}(r)$		Выход $\mathrm{Re}\,\hat{Z}_{N+1}(r)$	
	Re	Im	Re	Im
0	16	0	256	0
1	1	–10.15317039	–102.0868689	–20.30634078
2	0	0	0	0
3	1	–3.296558209	–9.867296025	–6.593116418
4	0	0	0	0
5	1	–1.870868412	–2.500148614	–3.741736824
6	0	0	0	0
7	1	–1.218503526	–0.4847508419	–2.437007051
8	0	0	0	0
9	1	–0.8206787908	0.3264863223	–1.641357582
10	0	0	0	0
11	1	–0.534511136	0.7142978455	–1.069022272
12	0	0	0	0
13	1	–0.3033466836	0.9079807895	–0.6066933672
14	0	0	0	0
15	1	–0.09849140336	0.9902994435	–0.1969828067
16	0	0	0	0

Таблица 7. Результат Алгоритма 9. Упаковка $\left(\hat{Z}_N(r)\right)$

r	Вход $\hat{Z}_{N+1}(r)$		Выход $\hat{Z}_N(r)$	
	Re	Im	Re	Im
0	256	0	128	128
1	–102.0868689	–20.30634078	–30.43892677	–58.60296372
2	0	0	0	0
3	–9.867296025	–6.593116418	1.506765433	–5.472869143
4	0	0	0	0
5	–2.500148614	–3.741736824	1.779789167	–0.2292826602
6	0	0	0	0
7	–0.4847508419	–2.437007051	0.7165172097	1.523043005
8	0	0	0	0
9	0.3264863223	–1.641357582	–0.8747817293	2.318692475
10	0	0	0	0
11	0.7142978455	–1.069022272	–3.565639936	2.443431891
12	0	0	0	0
13	0.9079807895	–0.6066933672	–10.46608067	0.5135539076
14	0	0	0	0
15	0.9902994435	–0.1969828067	–70.65764271	–38.49360575
16	0	0		

Таблица 8. Результат Алгоритма 10. Комплексное сопряжение $\left(\hat{Z}_N(r)\right)$

r	Вход $\hat{Z}_N(r)$		Выход $\hat{Z}_N(r)$	
	Re	Im	Re	Im
0	128	128	128	–128
1	–30.43892677	–58.60296372	–30.43892677	58.60296372
2	0	0	0	0
3	1.506765433	–5.472869143	1.506765433	5.472869143
4	0	0	0	0
5	1.779789167	–0.2292826602	1.779789167	0.2292826602
6	0	0	0	0
7	0.7165172097	1.523043005	0.7165172097	–1.523043005
8	0	0	0	0
9	–0.8747817293	2.318692475	–0.8747817293	–2.318692475
10	0	0	0	0
11	–3.565639936	2.443431891	–3.565639936	–2.443431891
12	0	0	0	0
13	–10.46608067	0.5135539076	–10.46608067	–0.5135539076
14	0	0	0	0
15	–70.65764271	–38.49360575	–70.65764271	38.49360575

Таблица 9. Результат Алгоритма 5. БПФ $\left(\hat{Z}_N\right)$

r	Вход $\hat{Z}_N(r)$		Выход $\hat{Z}_N(r)$	
	Re	Im	Re	Im
0	128	–128	16	–32
1	–30.43892677	58.60296372	240	–224
2	0	0	144	–160
3	1.506765433	5.472869143	122	–96
4	0	0	80	–96
5	1.779789167	0.2292826602	176	–160
6	0	0	208	–224
7	0.7165172097	–1.523043005	48	–32
8	0	0	48	–64
9	–0.8747817293	–2.318692475	208	–192
10	0	0	176	–192
11	–3.565639936	–2.443431891	80	–64
12	0	0	112	–128
13	–10.46608067	–0.5135539076	144	–128
14	0	0	240	–256
15	–70.65764271	38.49360575	16	0

Таблица 10. Результат Алгоритма 6. Битовая инверсия $\left(\hat{Z}_N(r)\right)$

r	Вход $\hat{Z}_N(r)$		Выход $\hat{Z}_N(r)$	
	Re	Im	Re	Im
0	16	–32	16	–32
1	240	–224	48	–64
2	144	–160	80	–96
3	122	–96	112	–128
4	80	–96	144	–160
5	176	–160	176	–192
6	208	–224	208	–224
7	48	–32	240	–256
8	48	–64	240	–224
9	208	–192	208	–192
10	176	–192	176	–160
11	80	–64	144	–128
12	112	–128	112	–96
13	144	–128	80	–64
14	240	–256	48	–32
15	16	0	16	0

Таблица 11. Результат Алгоритма 10.
Комплексное сопряжение ($\hat{Z}_N(r)$)

r	Вход $\hat{Z}_N(r)$		Выход $\hat{Z}_N(r)$	
	Re	Im	Re	Im
0	16	–32	16	32
1	240	–224	48	64
2	144	–160	80	96
3	122	–96	112	128
4	80	–96	144	160
5	176	–160	176	192
6	208	–224	208	224
7	48	–32	240	256
8	48	–64	240	224
9	208	–192	208	192
10	176	–192	176	160
11	80	–64	144	128
12	112	–128	112	96
13	144	–128	80	64
14	240	–256	48	32
15	16	0	16	0

Таблица 12. Результат Алгоритма 11.
Результат умножения ($R_{2N}(r)$)

r	Вход $\hat{Z}_N(r)$		Выход $\hat{Z}_N(r)$	
	Re	Im	$R_{2N}(2r)$	$R_{2N}(2r+1)$
0	16	32	1	2
1	48	64	3	4
2	80	96	5	6
3	112	128	7	8
4	144	160	9	10
5	176	192	11	12
6	208	224	13	14
7	240	256	15	16
8	240	224	15	14
9	208	192	13	12
10	176	160	11	10
11	144	128	9	8
12	112	96	7	6
13	80	64	5	4
14	48	32	3	2
15	16	0	1	0

Лемма 3. (Распаковка-Упаковка. Переход от большей разрядности $2N$ к меньшей N, используя фломулы распаковки). В условиях леммы 1 справедливы следующие соотношения:

$$\hat{Z}_N(r) \leftarrow \hat{Z}_N^*(r),\ r = \overline{0, N-1}. \quad (2)$$

$$\hat{X}_N(0) \leftarrow \frac{1}{2}\Big(\big(\mathrm{Re}\,\hat{Z}_N(0) + \mathrm{Re}\,\hat{Z}_N(N)\big) + i\big(\mathrm{Re}\,\hat{Z}_N(0) - \mathrm{Re}\,\hat{Z}_N(N)\big)\Big),$$

$$\hat{X}_N(N/2) \leftarrow \hat{Z}_N^*(N/2). \quad (3)$$

$$\hat{X}_N(r) \leftarrow A_{N/2}(r) + S_{N/2}(r),$$

$$\hat{X}_N(N-r) \leftarrow A_{N/2}^*(r) - S_{N/2}^*(r), \quad (4)$$

$$A_{N/2}(r) \leftarrow \frac{1}{2}\Big(\hat{Z}_N(r) + \hat{Z}_N^*(N-r)\Big),$$

$$S_{N/2}(r) \leftarrow \frac{W_{2N}^r}{2i}\Big(\hat{Z}_N(r) - \hat{Z}_N^*(N-r)\Big), \quad (5)$$

$$W_{2N}^r = e^{-\frac{\pi i}{N}r},\ r = \overline{1, N/2-1}. \quad (6)$$

$$\hat{X}_N(r) \leftarrow \hat{X}_N^*(r),\ r = \overline{0, N-1}. \quad (7)$$

*Доказательство***.** Видно, что формулы (3)–(6) являются формулами распаковки (лемма 1). Рассмотрим сначала выражения (2), (3), (7) для $r = 0$. Так как $\hat{Z}_N(0)$, $\hat{Z}_N(N)$ – действительные числа, то выполнение выражения (2) для $r = 0$ не оказывает влияния на результат $\hat{X}_N(0)$. При $r = N/2$ выражения $\hat{Z}_N(r) \leftarrow \hat{Z}_N^*(r)$, $\hat{X}_N(N/2) \leftarrow \hat{Z}_N^*(N/2)$, $\hat{X}_N(N/2) \leftarrow \hat{X}_N^*(N/2)$ дают результат $\hat{X}_N(N/2) \leftarrow \hat{Z}_N^*(N/2)$ с учетом свойства комплексных чисел $a \leftarrow \left(a^*\right)^*$.

Сделаем подстановку (2) в (5) и (4) в (7):

$$\hat{X}_N(r) = \left(A_{N/2}(r) + S_{N/2}(r)\right)^*,$$

$$\hat{X}_N(N-r) = \left(A_{N/2}^*(r) - S_{N/2}^*(r)\right)^*,$$

$$A_{N/2}(r) = \frac{1}{2}\left(\left(\hat{Z}_N(r)\right)^* + \left(\hat{Z}_N^*(N-r)\right)^*\right),$$

$$S_{N/2}(r) = \frac{W_{2N}^r}{2i}\left(\left(\hat{Z}_N(r)\right)^* - \left(\hat{Z}_N^*(N-r)\right)^*\right),$$

$$W_{2N}^r = e^{-\frac{\pi i}{N}r},\ r = \overline{1, N/2-1}. \quad (8)$$

С учетом свойств $f^* \pm g^* = (f \pm g)^*$, $(W_{2N}^{-r})^* = W_{2N}^r$, $f \cdot g^* = (f^* \cdot g)^*$, $\frac{f^*}{i} = -\left(\frac{f}{i}\right)^*$,

$\frac{1}{i} = -i$ где f и g – комплексные числа, выражения для $A_{N/2}(r)$ и $S_{N/2}(r)$ примут вид:

$$A_{N/2}(r) = \frac{1}{2}\left(\left(\hat{Z}_N(r)\right)^* + \left(\hat{Z}_N^*(N-r)\right)^*\right) = \frac{1}{2}\left(\hat{Z}_N(r) + \hat{Z}_N^*(N-r)\right)^*,$$

$$S_{N/2}(r) = \frac{W_{2N}^r}{2i}\left(\left(\hat{Z}_N(r)\right)^* - \left(\hat{Z}_N^*(N-r)\right)^*\right) = \frac{W_{2N}^r}{2i}\left(\hat{Z}_N(r) - \hat{Z}_N^*(N-r)\right)^* = -\left(\frac{W_{2N}^{-r}}{2i}\left(\hat{Z}_N(r) - \hat{Z}_N^*(N-r)\right)\right)^*.$$

Подставляя полученные выражения в (8) получим (1), что и требовалось доказать.

Свойство, доказанное выше, позволяет заменить формулы упаковки формулами распаковки при $r = \overline{1, N/2-1}$, что уменьшает число подпрограмм при реализации операции умножения больших чисел.

Интересно, что аналогичным образом можно избежать использования формул распаковки, заменив их формулами упаковки.

Алгоритм 12. Распаковка-Упаковка.

Вход: $\hat{X}_{N+M}(r)$, $r = \overline{0, N-1}$ $\left(\hat{X}_{N+1}(r),\ r = \overline{0, N}\right.$, при $\left.M = 1\right)$ – комплексный сигнал,

$C_N(r)$, $r = \overline{0, N-1}$ – вектор предвычисленных косинусов,

$B_N(r)$ – вектор номеров строк для битовой инверсии (см. Алгоритм 3. Предвычисления),

N – число разрядов комплексного сигнала.

M – режим вычисления (0 – распаковка, 1 – упаковка).

Выход: $\hat{X}_{N+1-M}$, $r = \overline{0, N}$, $\left(\hat{X}_N(r)\right.$, $\left.r = \overline{0, N-1}\right.$, при $\left.M = 1\right)$ – преобразованный ДПФ.

Шаг 1. Если $M = 0$, то

Шаг 2. $\hat{X}_{N+1}(0) \leftarrow \mathrm{Re}\, X + \mathrm{Im}\, X$, $\hat{X}_{N+1}(N) \leftarrow \mathrm{Re}\, X - \mathrm{Im}\, X$, $X \leftarrow \hat{X}_N(0)$.

Шаг 3. Иначе

Шаг 4.

$$\hat{X}_N(0) \leftarrow \frac{1}{2}\left(\left(\mathrm{Re}\,\hat{X}_{N+1}(0) + \mathrm{Re}\,\hat{X}_{N+1}(N)\right) + i\left(\mathrm{Re}\,\hat{X}_{N+1}(0) - \mathrm{Re}\,\hat{X}_{N+1}(N)\right)\right).$$

Шаг 5. Конец если.

Шаг 6. Для r от 1 до $N/2-1$.

Шаг 7. $k \leftarrow B_N(r)$;

$$W \leftarrow \begin{Bmatrix} \langle k \rangle_2 = 0, & C_N(k) + iC_N(k+1) \\ \langle k \rangle_2 = 1, & C_N(k) - iC_N(k-1) \end{Bmatrix}.$$

Шаг 8.

$$A \leftarrow \frac{1}{2}\left(\hat{X}_N(r) + \hat{X}_N^*(N-r)\right),$$

$$S \leftarrow \frac{1}{2}\left(\hat{X}_N(r) - \hat{X}_N^*(N-r)\right).$$

Шаг 9. $S \leftarrow (-i) \cdot W \cdot S$.

Шаг 10. $\hat{X}_N(r) \leftarrow A + S$,

$$\hat{X}_N(N-r) \leftarrow A^* - S^*.$$

Шаг 11. Конец цикла по r.

Тогда в Алгоритме 2 шаг 7–11 заменяются следующими выражениями:

Шаг 7. $\hat{X}_{N+1} \leftarrow$Распаковка-Упаковка($\hat{X}_N$, B_N, B_N, N, 0) (см. Алгоритм 12).

Шаг 8. $\hat{Y}_{N+1} \leftarrow$Распаковка-Упаковка ($\hat{Y}_N$, C_N, B_N, N, 0) (см. Алгоритм 12).

Шаг 9. $\hat{Z}_{N+1} \leftarrow$Умножение($\hat{X}_{N+1}$, $\hat{Y}_{N+1}$, N) (см. Алгоритм 8).

Шаг 10. $\hat{Z}_N \leftarrow$Комплексное сопряжение($\hat{Z}_N$, N) (см. Алгоритм 10).

Шаг 11. $\hat{Z}_N \leftarrow$Распаковка-Упаковка $\left(\hat{Z}_{N+1}, C_N, B_N, N, 1\right)$(см. Алгоритм 12).

Видим, что в оптимизированном алгоритме комплексное сопряжение выполняется сразу после поразрядного умножения сигналов.

Далее приведены реализации Алгоритма 2 и его оптимизация (язык программирования APL).

```
Rv←FFTMainF2;N;Uv;Uv;Cv;Bv;Xc;Yc;Zc
N←16
Uv←Uv←N⍴1

Cv Bv←FFTPreCalcF2 N
Xc Yc←FFTInitF2 Uv Uv N
Xc←FFTTransformF2 Xc Cv Bv N
Yc←FFTTransformF2 Yc Cv Bv N
Xc←FFTBinaryInverseF2 Xc Bv N
Yc←FFTBinaryInverseF2 Yc Bv N
Xc←FFTUnpackF2 Xc Cv Bv N
Yc←FFTUnpackF2 Yc Cv Bv N
Zc←FFTMultiF2 Xc Yc N
Zc←FFTPackF2 Zc Cv Bv N
Zc←FFTComplexF2 Zc N
Zc←FFTTransformF2 Zc Cv Bv N
Zc←FFTBinaryInverseF2 Zc Bv N
Zc←FFTComplexF2 Zc N
Rv←FFTSaveRes2 Zc N
```

```
Rv←FFTMainF2Opt;Uv;Uv;Xc;Yc;N;Cv;Bv;Zc
N←16
Uv←Uv←N⍴1

Cv Bv←FFTPreCalcF2 N
Xc Yc←FFTInitF2 Uv Uv N
Xc←FFTTransformF2 Xc Cv Bv N
Yc←FFTTransformF2 Yc Cv Bv N
Xc←FFTBinaryInverseF2 Xc Bv N
Yc←FFTBinaryInverseF2 Yc Bv N
Xc←FFTUnpackPackF2 Xc Cv Bv N 0
Yc←FFTUnpackPackF2 Yc Cv Bv N 0
Zc←FFTMultiF2 Xc Yc N
Zc←FFTComplexF2 Zc N
Zc←FFTUnpackPackF2 Zc Cv Bv N 1
Zc←FFTTransformF2 Zc Cv Bv N
Zc←FFTBinaryInverseF2 Zc Bv N
Zc←FFTComplexF2 Zc N
Rv←FFTSaveRes2 Zc N
```

```
Xc←FFTBinaryInverseF2 arg;Xc;Bv;N;f;r;j;X
⍝ Binary inversion
Xc Bv N←arg
f←1 ⍝ first element index
:For r :In 0,⍳N-1
    j←Bv[f+r]
    :If r<j
        X←Xc complGet j
        Xc←Xc complSet j(Xc complGet r)
        Xc←Xc complSet r X
    :EndIf
:EndFor
```

```
Xc←FFTComplexF2 arg;Xc;N;r;X
⍝ Change Sign
Xc N←arg

:For r :In 0,⍳N-1
    X←Xc complGet r
    Xc←Xc complSet r((Re X),(-Im X))
:EndFor
```

```
res←FFTPreCalcF2 N;Cv;Bv;f;r;n;j;d;p;k;v
Cv←Bv←N⍴0

f←1     ⍝ first element index
Bv[f+0]←0 ⋄ Cv[f+0]←cos(Bv[f+0])
j←1 ⋄ d←N÷2 ⋄ r←f ⋄ n←log2 N

:For p :In ⍳n             ⍝ p=1,2,..,N
    :For k :In 0,⍳j-1     ⍝ k=0,1,..,j-1
        v←Bv[f+k]+d
        Bv[f+r]←v ⋄ Cv[f+r]←cos(-(v÷2)×2×PI÷N)
        r←r+1
    :EndFor
    j←j×2 ⋄ d←d÷2
:EndFor

Cv[f+1]←0
res←Cv Bv
```

```
W←Cv GetW k;f;cosv;sinv
f←1 ⍝ first element index
:If 0=2|k
    cosv←Cv[f+k] ⋄ sinv←Cv[f+k+1]
:Else
    cosv←Cv[f+k] ⋄ sinv←-Cv[f+k-1]
:EndIf
W←cosv,sinv
```

```
Rv←FFTSaveRes2 arg;Zc;N;r;val;f;Z;K
Zc N←arg
K←2×N
Rv←K⍴0
f←1 ⍝ first element index
:For r :In 0,⍳N-1
    Z←Zc complGet r
    Rv[f+(r×2)]←(Re Z)÷N
    Rv[f+(r×2)+1]←(Im Z)÷N
:EndFor
Rv[K]←⌊Rv[K]+0.0000000001
```

```
Xc←FFTTransformF2 arg;Cv;Bv;N;S;j;d;n;p;k;s1;s2;r;i1;i2;X;W
Xc Cv Bv N←arg
S←(sqrt 2)÷2 ⋄ j←1 ⋄ d←N÷2 ⋄ n←log2 N

:For p :In 0,⍳n-1
    :For k :In 0,⍳j-1
        s1←k×(2×d) ⋄ s2←s1+d
        :For r :In 0,⍳d-1
            i1 i2←(s1 s2)+r
            X←Xc complGet i2
            :Select k
            :Case 0
                X←X                     ⍝ X×1
            :Case 1
                X←X complMul(0,-1)      ⍝ X×(-i)
            :Case 2
                X←X complMul(S,-S)      ⍝ X×(1-i)×sqrt 2  //-pi÷4
            :Case 3
                X←X complMul((-S),-S)   ⍝ X×-(1+i)×sqrt 2 //-3×pi÷4
            :Else
                W←Cv GetW k             ⍝ rad←-(Bv[k+1])×PI÷N
                X←X complMul W          ⍝ cosv←cos rad ⋄ sinv←sin rad
            :EndSelect                  ⍝ W←(cosv,sinv) ⋄ X←X×W

            Xc←Xc i2 i1 complSetGetSub X
            Xc←Xc i1 i1 complSetGetAdd X
        :EndFor
    :EndFor
    j←j×2 ⋄ d←d÷2
:EndFor
```

```
res←sqrt val
⍝ Square root
res←val*0.5
```

```
n←log2 N
n←2⍟N
```

```
res←cos val
res←2○val
```

```
Xc←Xc complSet arg;r;X;f
r X←arg
f←1 ⍝ first element index
Xc[f+(r×2)+0]←Re X
Xc[f+(r×2)+1]←Im X
```

```
Xr←Xc complGet r;f;re;im
f←1 ⍝ first element index
re←Xc[f+(r×2)+0]
im←Xc[f+(r×2)+1]
Xr←re,im
```

```
Xc←larg complSetGetSub T;i1;i2
Xc i2 i1←larg
T←(Xc complGet i1)-T
Xc←Xc complSet i2 T
```

```
Xc←larg complSetGetAdd T;i1;i2
Xc i2 i1←larg
T←(Xc complGet i1)+T
Xc←Xc complSet i2 T
```

```
TRes←T1 complMul T2;T1re;T1im;T2re;T2im;TResre;TResim
T1re T1im←T1
T2re T2im←T2

TResre←(T1re×T2re)-(T1im×T2im)
TResim←(T1im×T2re)+(T1re×T2im)
TRes←TResre,TResim
```

```
res←FFTInitF2 arg;Uv;Uv;N;Xc;Yc
Uv Uv N←arg
Xc←((N+1)×2)⍴0    ⍝ N+1 length
Yc←((N+1)×2)⍴0    ⍝ N+1 length
Xc[⍳N]←Uv[⍳N]
Yc[⍳N]←Uv[⍳N]
res←Xc Yc
```

```
Xc←FFTUnpackPackF2 arg;Xc;Cv;Bv;N;X;mode;f;XOre;XNre;re;im;r
⍝ Transfrom from N <-> 2N
Xc Cv Bv N mode←arg
f←1 ⍝ first element index

:If mode=0
    X←Xc complGet 0
    Xc←Xc complSet 0(((Re X)+(Im X)),0)
    Xc←Xc complSet N(((Re X)-(Im X)),0)
:Else
    XOre←Xc complGet 0
    XNre←Xc complGet N
    re←((Re XOre)+(Re XNre))÷2
    im←((Re XOre)-(Re XNre))÷2
    Xc←Xc complSet 0(complComlex(re,im))
:EndIf

⍝Xc←Xc complSet(N÷2)(complComlex(Xc complGet(N÷2)))

:For r :In ⍳(N÷2)-1
    k←Bv[f+r]           ⍝ ... K←2×N ○ rad←-(2×PI×r)÷K
    W←Cv GetW k         ⍝ ... cosv←cos rad ○ sinv←sin rad
                        ⍝ ... W←cosv,sinv
    Xr←Xc complGet r
    XNr←Xc complGet(N-r)
    XNrC←complComplex XNr

    A←(Xr+XNrC)÷2
    S←((Xr-XNrC)÷2)complMul(0,-1)          ⍝ 1÷i=-i
    S←W complMul S

    Xc←Xc complSet r(A+S)
    Xc←Xc complSet(N-r)(complComplex A-S)
:EndFor
```

```
Zc←FFTMultiF2 arg;Xc;Yc;N;XOre;YOre;XNre;YNre;
Xc Yc N←arg
Zc←((N+1)×2)⍴0

XOre←Re Xc complGet 0
YOre←Re Yc complGet 0
Zc←Zc complSet 0(XOre×YOre,0)

XNre←Re Xc complGet N
YNre←Re Yc complGet N
Zc←Zc complSet N(XNre×YNre,0)

:For r :In 0,⍳N-1
    Xr←Xc complGet r
    Yr←Yc complGet r
    Zc←Zc complSet r(Xr complMul Yr)
:EndFor
```

```
Xc←FFTUnpackF2 arg;Cv;Bv;N;f;X;r;k;W;Xr;XNr;XNrC;A;S
⍝ Transfrom from N to 2N
Xc Cv Bv N←arg
f←1 ⍝ first element index

X←Xc complGet 0
Xc←Xc complSet 0(((Re X)+(Im X)),0)
Xc←Xc complSet N(((Re X)-(Im X)),0)

:For r :In ⍳(N÷2)-1
    k←Bv[f+r]       ⍝ ... K←2×N ○ rad←-(2×PI×r)÷K
    W←Cv GetW k     ⍝ ... cosv←cos rad ○ sinv←sin rad
                    ⍝ ... W←cosv,sinv
    Xr←Xc complGet r
    XNr←Xc complGet(N-r)
    XNrC←complComplex XNr

    A←(Xr+XNrC)÷2
    S←((Xr-XNrC)÷2)complMul(0,-1)          ⍝ 1÷i=-i
    S←W complMul S

    Xc←Xc complSet r(A+S)
    Xc←Xc complSet(N-r)(complComplex A-S)
:EndFor
```

```
Zc←FFTPackF2 arg;Zc;Cv;Bv;N;f;ZOre;ZNre;re;im;
Zc Cv Bv N←arg
f←1 ⍝ first element index

ZOre←Zc complGet 0
ZNre←Zc complGet N
re←((Re ZOre)+(Re ZNre))÷2
im←((Re ZOre)-(Re ZNre))÷2
Zc←Zc complSet 0(re,im)

:For r :In ⍳(N÷2)-1
    k←Bv[f+r]           ⍝ ... K←2×N ○ rad←-(2×P
    W←Cv GetW k         ⍝ ... cosv←cos rad ○ si
                        ⍝ ... W←cosv,sinv
    Zr←Zc complGet r
    ZNr←Zc complGet(N-r)
    ZNrC←complComplex ZNr

    A←(Zr+ZNrC)÷2
    S←((Zr-ZNrC)÷2)complMul(0,-1)
    S←(complComplex W)complMul S

    Zc←Zc complSet r(A-S)
    Zc←Zc complSet(N-r)(complComplex A+S)
:EndFor
```

Выполнение программ **FFTMainF2** и **FFTMainF2Opt** дают одинаковый результат:

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0[2]

Младший разряд слева.

Вывод

Предложена оптимизация алгоритма умножения больших N-разрядных чисел на основе N-разрядных БПФ с использованием формул распаковки-упаковки [6]. Оптимизация позволяет уменьшить число модулей (подпрограмм), что уменьшает общую сложность алгоритма умножения. Показан метод замены формул упаковки формулами распаковки. Приведено детальное описание алгоритма реализации операции умножения.

1. *Задірака В., Олексюк О.* Комп'ютерна арифметика багаторозрядних чисел. – К.: Наук. думка, 2003. – 263 с.
2. *Schonhage A., Strassen V.* Schnelle Multiplikation grossen Zahnel // Computing. – 1971. – 7, N 3–4. – P. 281–292.
3. *Шенхаге А., Шрассен В.* Быстрое умножение больших чисел // Кибернетика. – 1972. – Вып. 2. – С. 87–98.
4. *Cooley J.W., Tukey J.W.* An algorithm for the machine calculation of complex Fourier Series // Math Compt. – 1965. Apr. – P. 257–301.
5. *Березовский А.И., Задирака В.К., Шевчук Л.Б.* О тестировании быстродействия алгоритмов и программ выполнения основных операций для асимметричной криптографии // Кибернетика и системный анализ. – 1999. – № 5. – С. 61–68.
6. *Терещенко А.Н.* Умножение больших N-разрядных чисел с вычислением только N-разрядных ДПФ // Компьютерная математика. – 2008. – № 1. – С. 122–130.

Получено 13.08.2012

Об авторах:

Терещенко Андрей Николаевич,
кандидат физико-математических наук,
начальник отдела по поддержке информационных систем,

Задирака Валерий Константинович,
доктор физико-математических наук,
профессор, член-корреспондент НАН Украины,
заведующий отделом.

Место работы авторов:

ООО «Симкорп Украина»,
Киев, ул. В. Стусса 35-37,
teramidi@ukr.net

Институт кибернетики
имени В.М. Глушкова НАН Украины,
03680, Київ-187,
проспект Академика Глушкова, 40,
(044) 526 4568,
zvk140@ukr.net

[2] Каждый разряд – 1 байт