УДК 621.374.325.01

К. ХЕРРМАНН
(Львов)

МИНИМИЗАЦИЯ ВРЕМЕНИ ВВОДА — ВЫВОДА ПРИ ФУРЬЕ-ОБРАБОТКЕ БОЛЬШИХ МАССИВОВ ДАННЫХ

В данной работе синтезируются алгоритмы быстрого преобразования Фурье (БПФ) для преобразования больших массивов данных, хранящихся на дисковых запоминающих устройствах (ДЗУ). Известна модификация алгоритма Кули — Таки с основанием $p = 2$, предложенная Синглтоном [1], которая позволяет сэкономить время ввода-вывода за счет преобразования независимых промежуточных векторов, имеющих размер, равный объему оперативной памяти (без обращения к ДЗУ). В [2] предлагается выполнить фурье-преобразование за две итерации, однако такой подход приводит к существенным затратам машинного времени из-за необходимости транспонирования всего массива данных, и, кроме того, для его реализации требуется дополнительная память.

Можно синтезировать алгоритм БПФ с основанием $p = 2^\varepsilon$ $(\varepsilon > 1)$, не требующий транспонирования массива данных [3]. Наиболее простые алгоритмы получаются в случае $\varepsilon = \gamma - m$ (2^γ — размер оперативной памяти, 2^m — блока ДЗУ).

Ниже исследуются алгоритмы БПФ с основанием $p = 2^\varepsilon$ $(\varepsilon \geqslant 1)$ в зависимости от таких параметров ДЗУ, как время доступа и считывания (записи) слова с (на) ДЗУ, размер блока ДЗУ. Найдена зависимость основания алгоритма БПФ от размера блока ДЗУ, которая позволяет синтезировать алгоритм БПФ с минимальным временем ввода-вывода.

Пусть размер массива данных $N = 2^k$ превышает размер оперативной памяти $N_0 = 2^\gamma$. При алгоритме с основанием $p = 2^\varepsilon$ исходный массив представляется $2^{k-\gamma+\varepsilon}$ блоками размером $2^{\gamma-\varepsilon}$, поэтому необходимо выполнить $\left[\log_{2^\varepsilon} 2^{k-\gamma+\varepsilon}\right] = \left[\frac{k-\gamma+\varepsilon}{\varepsilon}\right]$ ([ ] — целая часть) требующих обращения к ДЗУ итераций по основанию 2^ε и одну итерацию по основанию 2^T, где $T = k - \gamma + \varepsilon - \varepsilon\left[\frac{k-\gamma+\varepsilon}{\varepsilon}\right]$. При $T = 0$ необходимость в итерации по основанию 2^T отпадает.

Предположим, что при обращении к ДЗУ вводится (выводится) один блок размером 2^m и время доступа (обращения) к этому блоку не зависит от размера оперативной памяти, основания алгоритма, номера итерации и номера блока в массиве данных. При принятых предположениях время ввода (вывода) блока размером 2^m равно $T_д + 2^m T_п$, где $T_п$ — время передачи слова ДЗУ и $T_д$ — время доступа.

При $\gamma - \varepsilon \geqslant m$ число блоков, извлекаемых на отдельной итерации, не зависит от основания и равно 2^{k-m}. При этом время ввода (вывода)

$$T_1 = \begin{cases} 2^k\left(\left[\frac{k-\gamma+\varepsilon}{\varepsilon}\right]+1\right)\left(2^{-m}T_д + T_п\right) & \text{при } T > 0, \\ 2^k\frac{k-\gamma+\varepsilon}{\varepsilon}\left(2^{-m}T_д + T_п\right) & \text{при } T = 0. \end{cases} \qquad (1)$$

Из (1) видно, что при $2^{-m}T_д/T_п \ll 1$ минимальное значение $T_{1\min} = 2^{k+1}(2^{-m}T_д + T_п)$ в случае, если $\varepsilon = \begin{cases}\gamma & \text{при } \gamma \geqslant k/2, \\ k-\gamma & \text{при } \gamma \leqslant k/2.\end{cases}$

На последней итерации при $\gamma - \varepsilon < m$ время ввода (вывода) получается минимальным, когда извлекаются блоки размером 2^m. Для остальных итераций полагаем, что за одно обращение к ДЗУ извлекается не более одного блока размером $2^{\gamma-\varepsilon} < 2^m$.

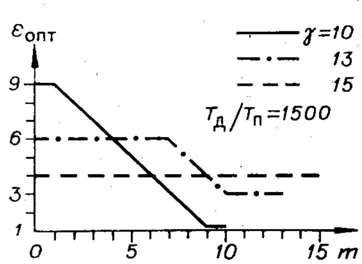

Рис. 1. Зависимость оптимального значения ε от параметра m для $k = 19$.

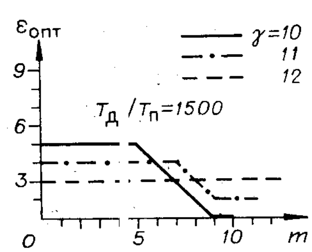

Рис. 2. Зависимость оптимального значения ε от параметра m для $k = 15$.

В случае, когда $2^{k-\gamma}T_д \gg 2^{-m}T_д \gg T_п$, минимальное значение $T_{2\min} \approx$ $\approx 2^k(k-\gamma)(T_д/2^{\gamma-1}+T_п)$ достигается при $\varepsilon = 1$.

На рис. 1—2 показаны оптимальные значения ε при некоторых типичных k, γ, $T_д/T_п$. Из рис. 1—2 видно, что при $m \leqslant 2\gamma - k$ оптимальное значение приближенно равно $\varepsilon_{опт} = k - \gamma$, так что наиболее экономичными по времени ввода-вывода являются алгоритмы Грейнджера [2], выполняемые в нашем случае без транспонирования массива данных.

При $\alpha(k-\gamma,\ T_д/T_п) > m > 2\gamma - k$ оптимальное значение ε приближенно равно $\varepsilon_{опт} = \gamma - m$. Таким образом, наиболее рациональным в этом случае является алгоритм Фрейзера [3].

На рис. 1—2 показано, где существуют такие значения $m > \alpha(k-\gamma,\ T_д/T_п)$, при которых предложенные в [2] и [3] алгоритмы неоптимальны, а $\gamma - m < \varepsilon_{опт} < k - \gamma$.

ЛИТЕРАТУРА

1. **Singleton R. C.** On Computing the Fast Fourier Transform.— Comm. of the ACM, 1967, vol. 10, N 10.
2. **Brenner N. M.** Fast Fourier Transform of Externally Stored Data.— IEEE Trans. on Audio and Electroac., 1969, vol. AU-17, N 2.
3. **Fraser D.** Optimized Mass Storage FFT Program.— In: Programs for Digital Signal Processing. The IEEE, Inc., N.-Y., 1979.

Поступило в редакцию 7 сентября 1981 г.

ВНИМАНИЮ ЧИТАТЕЛЕЙ!

В № 6 за 1982 г. замечены следующие опечатки:

Страница	Строка	Напечатано	Следует читать
56	9-я (сверху)	$d/dc(\varepsilon^2)\|_{c=0} = 0.$	$\partial/\partial c(\varepsilon^2)\|_{c=0} = 0.$
78	29-я (снизу)	заполненных	заполненных
82	9-я (сверху)	$b_{\alpha'}$	$b_{\alpha'}$,
85	6-я (снизу)	обычной	обычный