Mathematical Problems of Computer Science 48, 23–32, 2017.

About Complexity of FFT Algorithms for Length of $q \times 2^p$

Rafayel V. Barseghyan

Institute for Informatics and Automation Problems of NAS RA
e-mail: rafayelbarseghyan@ipia.sci.am

Abstract

The paper presents logarithmic formula which allows to compute the exact number of necessary operations for computing the discrete Fourier transform (DFT) of an arbitrary $q \times 2^p$ - length, where q is an odd integer.

Keywords: Fast Fourier transform (FFT), Split-radix algorithm, Computational complexity.

1. Introduction

The discrete Fourier transform(DFT) has a wide range of applications in many fields of science and engineering[1],[2]. The main reason for its popularity is the existence of various algorithms which allows to significantly reduce the computational complexity. These algorithms are generally known as fast Fourier transforms (FFT). Fast algorithm for efficient computation of DFT was first introduced by Culey and Tukey by their historical paper in 1965 [3]. FFT algorithms allow to compute DFT of size N with $O(N \lg N)$ operations in contrast to direct form computation which requires $O(N^2)$ operations. There are a number of FFT algorithms, but the most popular methods are based on fixed-radix and split-radix approaches. Split-radix algorithms have been considered to be the most computationally efficient and structurally regular.

Split-radix algorithm was first introduced by Yavne [4] in 1969 and later by various authors [8]. Split-radix algorithm allows to compute DFT of $N = 2^p$ with $4N \lg N - 6N + 8$ arithmetic operations. In recent years by various authors [5], a new modification of split-radix algorithm was developed which allows to perform DFT of $N = 2^p$ with $\frac{34}{9} N \lg N - \frac{124}{27} N - 2 \lg N - \frac{2}{9}(-1)^{\lg N} \lg N + \frac{16}{27}(-1)^{\lg N} + 8$ arithmetic operations.

For applications, which need to perform DFT of sizes $N \neq 2^p$, usually specialists use the zero padding technique. It means that the input sequence is filled with zeroes until it becomes a power of two length for performing any available FFT algorithm. Such method significantly decreases the required number of arithmetic operations. DFT for input sequences which has length non-power-of-2 is required in many practical applications, it is an important problem.

Algorithm for computing DFT for sizes $q \times 2^p$, where q is an odd integer, was introduced by Bi and Chen in 1998 [6]. Algorithm has a 2/4 split-radix structure and in case of $q = 1$ has the same complexity as the conventional split-radix FFT algorithm. After that, in 2004 by Bouguezel and et. al. [10] a new improved algorithm for $q \times 2^p$ length DFT was presented.

Algorithm is based on 2/8 split-radix FFT algorithm scheme and improves such important factors as data transfer, address generation, twiddle factor computation and access to the lookup table, but it did not reduce number of arithmetic operations. In 2010 Bi and Chen [7] published a new paper where they presented a unified method for generation of $2/2a$ (where a is an integer and $a > 1$) split-radix algorithms for $q \times 2^p$ length DFTs.

In this paper a general logarithmic formula is derived for calculating number of arithmetic operations for 2/4 and 2/8 split-radix algorithms for $q \times 2^p$ length DFTs [14]. For all $q < 20$ special cases, formulas are developed for counting exact number of arithmetic operations and some cases are shown, where computational effectiveness is inversely proportional to the length of the DFT-size.

2. General Algorithm

Let $x = \{x_0, x_1, \ldots, x_{N-1}\}^T$ be a complex valued column-vector of length N, where $N = q \times 2^p$ and q is an odd integer. The DFT of this vector are defined as

$$X[k] = \sum_{k=0}^{N-1} x[n] W_N^{nk}, \tag{1}$$

where

$$0 \leq k \leq N-1, \quad W_N^n = \exp(-j\tfrac{2\pi}{N}n) = \cos(\tfrac{2\pi}{N}n) - j\sin(\tfrac{2\pi}{N}n), \quad j = \sqrt{-1}.$$

Below the algorithm from [7] is presented. Even indexes of the transform are computed by

$$X[2k] = \sum_{n=0}^{N/2-1} (x[n] + x[n+N/2]]) W_{N/2}^{nk}, \tag{2}$$

where $X[2k]$ is an $N/2$ DFT. The odd indexes are defined by

$$X[2ak+l] = \sum_{n=0}^{N-1} x[n] W_N^{n(2ak+l)}, \tag{3}$$

where $0 \leq k \leq (N/2a) - 1$, and a is an integer ($a > 1$) and l has a selected odd values so that $2ak + l$ generates $N/2$ odd integers that can be uniquely matched to all the odd index values between 0 and N. With some manipulations based on the periodic and symmetric properties of $W_N^{n(2ak+l)}$, (3) can be represented as

$$\begin{aligned} X[2ak+l] = &\sum_{n=0}^{(N/2a)-1} x'[n] W_N^{nl} W_{N/2a}^{nk} + \sum_{n=0}^{(N/2a)-1} x'[n+N/2a] W_N^{(n+N/2a)l} W_{N/2a}^{nk} + \ldots \\ &+ \sum_{n=0}^{(N/2a)-1} x'[n + \frac{(a-1)N}{2a}] W_N^{(n+\frac{(a-1)N}{2a})l} W_{N/2a}^{nk}, \end{aligned} \tag{4}$$

where $n = 0, 1, 2 \ldots, N/2 - 1$ and

$$x'[n] = x[n] - x[n + N/2]. \tag{5}$$

It can be observed that (4) is a length-$N/2a$ DFT the input sequence of which is the result of the computation inside the brackets of (4) for $n = 0, 1, 2 \ldots, N/2 - 1$. In summary, the even indexed outputs of (1) are obtained from one length-$N/2$ DFT defined in (2) based on

the radix-2 decomposition, and the odd indexed outputs are obtained from a length-$N/2a$ DFTs, based on the radix-$2a$ decomposition. The complexity of algorithm can be computed by the following expressions:

$$\begin{aligned} C_N^{\times} &= C_{N/2}^{\times} + aC_{N/2a}^{\times} + \tfrac{N}{2a}C^{\times} + 2N - C_t^{\times}, \\ C_N^{+} &= C_{N/2}^{+} + aC_{N/2a}^{+} + \tfrac{N}{2a}C^{+} + 3N - C_t^{+}, \end{aligned} \tag{6}$$

where by $C^{\times}$ and C^{+} denote the number of real multiplications and additions that are used for each of the inner sums defined in (4), and $C_t^{\times}$ and C_t^{+} is the number of real multiplications and additions saved from all the trivial twiddle factors W_N^{nl} in (4).

3. 2/4 Split-Radix Algorithm

For $a = 2$ the algorithm becomes a modified version of conventional 2/4 split-radix algorithm. Inserting $a = 2$ into (4) we get

$$\begin{aligned} X[4k+l] &= \sum_{n=0}^{N/4-1} W_N^{nl}\Big(\sum_{n=0}^{1} x'[n+i\frac{N}{4}]W_{N/4}^{il}\Big)W_{N/4}^{nk} = \\ &= \sum_{n=0}^{N/4-1} W_N^{nl}(x'[n] + (-j)^l x'[n+\frac{N}{4}])W_{N/4}^{nk}, \end{aligned} \tag{7}$$

From (7) we can see that it becomes a conventional split-radix algorithm which is reported in [4],[8],[9]. To cover all odd indexes we set $l = \{-1, 1\}$. In this case, we have an arithmetic computational gain only in cases of $n = 0$ and $n = N/8$ (W_N^0 and $W_N^{lN/8}$ twiddle factors become trivial).
Now it is easy to see that the number of arithmetic operations are

$$\begin{aligned} C_N^{+} &= C_{N/2}^{+} + 2C_{N/4}^{+} + 4N - 4q \\ C_N^{\times} &= C_{N/2}^{\times} + 2C_{N/4}^{\times} + 2N - 12q. \end{aligned} \tag{8}$$

Using the theory of difference equations and Maxima [12] computer algebra system, we get the number of arithmetic operations required for computation of (7) in logarithmic form

$$\begin{aligned} C_N^{+} &= -\tfrac{2^p(28q-3C_{2q}^{+}-3C_q^{+})}{9} + \tfrac{(-1)^p(10q-3C_{2q}^{+}+6C_q^{+})}{9} + \tfrac{p2^{p+3}q}{3} + 2q, \\ C_N^{\times} &= -\tfrac{2^p(44q-3C_{2q}^{\times}-3C_q^{\times})}{9} - \tfrac{(-1)^p(10q+3C_{2q}^{\times}-6C_q^{\times})}{9} + \tfrac{p2^{p+2}q}{3} + 6q, \end{aligned} \tag{9}$$

where C_q and C_{2q} denote the complexities of q and $2q$ length DFTs, respectively. Using methods from [6] for computing $2q$-length DFT we obtain

$$\begin{aligned} C_N^{+} &= 2C_q^{+} + 4q, \\ C_N^{\times} &= 2C_q^{\times}, \end{aligned} \tag{10}$$

finally substituting (10) into (9) and $2^p = \frac{N}{q}$ we get

$$\begin{aligned} C_N^{+} &= \tfrac{8}{3}pq2^p - \tfrac{2^p}{9}(16q - 9C_q^{+}) - \tfrac{2}{9}q(-1)^p + 2q, \\ C_N^{\times} &= \tfrac{4}{3}pq2^p - \tfrac{2^p}{9}(44q - 9C_q^{\times}) - \tfrac{10}{9}q(-1)^p + 6q, \end{aligned} \tag{11}$$

$4q$-length DFT can be computed by

$$C_{4q}^{+} = 4C_q^{+} + 16q$$
$$C_{4q}^{\times} = 4C_q^{\times}$$

Using it and (8), finally we get the formulas which show the number of real arithmetic operations of $q \times 2^p$

$$\begin{aligned} C_N^{+} &= \tfrac{8}{3}pq2^p - \tfrac{2^p}{9}(16q - 9C_q^{+}) - \tfrac{2}{9}q(-1)^p + 2q =, \\ &= \tfrac{8}{3}N\log_2\left(\tfrac{N}{q}\right) - N\left(\tfrac{16}{9} - \tfrac{1}{q}C_q^{+}\right) - \tfrac{2}{9}q(-1)^{\log_2\left(\frac{N}{q}\right)} + 2q, \\ C_N^{\times} &= \tfrac{4}{3}pq2^p - \tfrac{2^p}{9}(38q - 9C_q^{\times}) + \tfrac{2}{9}q(-1)^p + 6q =, \\ &= \tfrac{4}{3}N\log_2\left(\tfrac{N}{q}\right) - N\left(\tfrac{38}{9} - \tfrac{1}{q}C_q^{+}\right) + \tfrac{2}{9}q(-1)^{\log_2\left(\frac{N}{q}\right)} + 6q. \end{aligned} \tag{12}$$

If $q = 1$ and, therefore $C_1^{+} = 0$ and $C_1^{+} = 0$ from (12) we can get

$$\begin{aligned} C_N^{+} &= \tfrac{8}{3}N\log_2 N - \tfrac{16}{9}N - \tfrac{2}{9}(-1)^{\log_2 N} + 2, \\ C_N^{\times} &= \tfrac{8}{3}N\log_2 N - \tfrac{38}{9}N + \tfrac{2}{9}(-1)^{\log_2 N} + 6. \end{aligned} \tag{13}$$

(13) is the same as the number of real arithmetic operations count required by conventional split-radix algorithm. Doing some optimization from [6], we get the following recurrent expressions for computing $8q$ length DFTs.

$$\begin{aligned} C_N^{+} &= C_{4q}^{+} + 4C_q^{+} + 36q = 8C_q^{+} + 52q, \\ C_N^{\times} &= C_{4q}^{\times} + 2C_q^{\times} + 2C_{sq}^{\times} = 6C_q^{\times} + 2C_{sq}^{\times}, \end{aligned} \tag{14}$$

where $C_{sq}^{\times}$ denotes the number of arithmetic operations required by scaled DFT [6]. Using (14), we get an improvement in the number of arithmetic operations

$$\begin{aligned} C_N^{+} &= \tfrac{8}{3}pq2^p - \tfrac{2^p}{9}(16q - 9C_q^{+}) - \tfrac{2}{9}q(-1)^p + 2q =, \\ &= \tfrac{8}{3}N\log_2\left(\tfrac{N}{q}\right) - N\left(\tfrac{16}{9} - \tfrac{1}{q}C_q^{+}\right) - \tfrac{2}{9}q(-1)^{\log_2\left(\frac{N}{q}\right)} + 2q, \\ C_N^{\times} &= \tfrac{4}{3}pq2^p - \tfrac{2^p}{18}(82q - 3(5C_q^{\times} + C_{sq}^{\times})) + \tfrac{2}{9}(7\,q + 3\,(C_q^{\times} - C_{sq}^{\times}))(-1)^p + 6q =, \\ &= \tfrac{4}{3}N\log_2\left(\tfrac{N}{q}\right) - \tfrac{N}{18}\left(82 - \tfrac{3}{q}(5C_q^{\times} + C_{sq}^{\times})\right) + \tfrac{2}{9}(7q + 3(C_q^{\times} - C_{sq}^{\times}))(-1)^{\log_2\left(\frac{N}{q}\right)} + 6q. \end{aligned} \tag{15}$$

3.1 Complexity of 2/4 Split-Radix Algorithm for $q < 20$

In [6] a method for computing 3-point DFT with optimization in case of 24-point is presented. Using that result we can get a complete expression which describes the number of arithmetic operations required by 3×2^p-length DFT.

$$\begin{aligned} C_N^{+}(3 \times 2^p) &= 8p2^p + \tfrac{20}{3}2^p - \tfrac{2}{3}(-1)^p + 6 = \\ &= \tfrac{8}{3}N\log_2(\tfrac{N}{3}) + \tfrac{20}{9}N - \tfrac{2}{3}(-1)^{\log_2(\frac{N}{3})} + 6, \\ C_N^{\times}(3 \times 2^p) &= 4\,p\,2^p - 11\;2^p + 2\,(-1)^p + 18 = \\ &= \tfrac{4}{3}N\log_2(\tfrac{N}{3}) - \tfrac{11}{3}N + 2(-1)^{\log_2(\frac{N}{3})} + 18. \end{aligned} \tag{16}$$

For $q = 9$

$$\begin{aligned} C_N^+(9 \times 2^p) &= 24\,p\,2^p + 68\;2^p - 2\,(-1)^p + 18 = \\ &= \tfrac{8}{3} N \log_2\left(\tfrac{N}{9}\right) - \tfrac{68}{81} N - \tfrac{2}{3}(-1)^{\log_2\left(\frac{N}{9}\right)} + 18, \\ C_N^\times\,(9 \times 2^p) &= 12\,p\,2^p - 24\;2^p + 10\,(-1)^p + 54 = \\ &= \tfrac{4}{3} N \log_2\left(\tfrac{N}{9}\right) - \tfrac{8}{3} N + 10\,(-1)^{\log_2\left(\frac{N}{9}\right)} + 54. \end{aligned} \tag{17}$$

For $q = 15$

$$\begin{aligned} C_N^+(15 \times 2^p) &= 40\,p\,2^p + \tfrac{424}{3} 2^p - \tfrac{10}{3}(-1)^p + 30 = \\ &= \tfrac{8}{3} N \log_2\left(\tfrac{N}{15}\right) - \tfrac{424}{45} N - \tfrac{10}{3}(-1)^{\log_2\left(\frac{N}{15}\right)} + 30, \\ C_N^\times\,(15 \times 2^p) &= 20\,p\,2^p - \tfrac{109}{3} 2^p + \tfrac{46}{3}(-1)^p + 90 = \\ &= \tfrac{4}{3} N \log_2\left(\tfrac{N}{15}\right) - \tfrac{109}{45} N - \tfrac{2}{3}(-1)^{\log_2\left(\frac{N}{15}\right)} + 90. \end{aligned} \tag{18}$$

For computing $q = 7$ length DFT, we use the method presented in [13]. That method allows to compute DFT with $C_7^+ = 72,\ C_7^\times = 16$. For getting a full logarithmic expression for 7×2^p, we use (12).

$$\begin{aligned} C_N^+(7 \times 2^p) &= \tfrac{7}{3} 2^{3+p} p + \tfrac{67}{9} 2^{3+p} - \tfrac{14}{9}(-1)^p + 14 = \\ &= \tfrac{8}{3} N \log_2\left(\tfrac{N}{7}\right) + \tfrac{536}{63} N - \tfrac{14}{9}(-1)^{\log_2\left(\frac{N}{7}\right)} + 14, \\ C_N^\times(7 \times 2^p) &= \tfrac{7}{3} 2^{2+p} p - \tfrac{61}{18} 2^p + \tfrac{14}{9}(-1)^p + 42 = \\ &= \tfrac{4}{3} N \log_2\left(\tfrac{N}{7}\right) - \tfrac{61}{126} N + \tfrac{14}{9}(-1)^{\log_2\left(\frac{N}{7}\right)} + 42. \end{aligned} \tag{19}$$

In case of $q = 11$ from [13], we have $C_{11}^+ = 168,\ C_{11}^\times = 40$

$$\begin{aligned} C_N^+(11 \times 2^p) &= \tfrac{11}{3} 2^{3+p} p + \tfrac{167}{9} 2^{3+p} - \tfrac{22}{9}(-1)^p + 22 = \\ &= \tfrac{8}{3} N \log_2\left(\tfrac{N}{11}\right) + \tfrac{1336}{99} N - \tfrac{22}{9}(-1)^{\log_2\left(\frac{N}{11}\right)} + 22, \\ C_N^\times(11 \times 2^p) &= \tfrac{11}{3} 2^{2+p} p - \tfrac{29}{9} 2^{1+p} + \tfrac{22}{9}(-1)^p + 66 = \\ &= \tfrac{4}{3} N \log_2\left(\tfrac{N}{11}\right) - \tfrac{58}{99} N + \tfrac{22}{9}(-1)^{\log_2\left(\frac{N}{7}\right)} + 66. \end{aligned} \tag{20}$$

In case of $q = 13$ from [13], we have $C_{13}^+ = 188,\ C_{13}^\times = 40$

$$\begin{aligned} C_N^+(13 \times 2^p) &= \tfrac{13}{3} 2^{3+p} p + \tfrac{371}{9} 2^{2+p} - \tfrac{26}{9}(-1)^p + 26 = \\ &= \tfrac{8}{3} N \log_2\left(\tfrac{N}{13}\right) + \tfrac{1484}{117} N - \tfrac{26}{9}(-1)^{\log_2\left(\frac{N}{13}\right)} + 26, \\ C_N^\times(13 \times 2^p) &= \tfrac{13}{3} 2^{2+p} p - \tfrac{67}{9} 2^{1+p} + \tfrac{26}{9}(-1)^p + 78 = \\ &\tfrac{4}{3} N \log_2\left(\tfrac{N}{7}\right) - \tfrac{134}{117} N + \tfrac{26}{9}(-1)^{\log_2\left(\frac{N}{7}\right)} + 78. \end{aligned} \tag{21}$$

In case of $q = 17$ from [13], we have $C_{17}^{+} = 274,\ C_{13}^{\times} = 82$

$$\begin{aligned} C_N^{+}(17 \times 2^p) &= \tfrac{17}{3} 2^{3+p} p + \tfrac{1097}{9} 2^{1+p} - \tfrac{34}{9}(-1)^p + 34 = \\ &= \tfrac{8}{3} N \log_2\left(\tfrac{N}{17}\right) + \tfrac{2194}{153} N - \tfrac{34}{9}(-1)^{\log_2\left(\frac{N}{17}\right)} + 34, \\ C_N^{\times}(17 \times 2^p) &= \tfrac{17}{3} 2^{2+p} p + \tfrac{23}{9} 2^{2+p} + \tfrac{34}{9}(-1)^p + 102 = \\ &= \tfrac{4}{3} N \log_2\left(\tfrac{N}{17}\right) - \tfrac{92}{153} N + \tfrac{34}{9}(-1)^{\log_2\left(\frac{N}{17}\right)} + 102. \end{aligned} \tag{22}$$

In case of $q = 19$ from [13], we have $C_{19}^{+} = 404,\ C_{19}^{\times} = 76$

$$\begin{aligned} C_N^{+}(19 \times 2^p) &= \tfrac{19}{3} 2^{3+p} p + \tfrac{833}{9} 2^{2+p} - \tfrac{38}{9}(-1)^p + 38 = \\ &= \tfrac{8}{3} N \log_2\left(\tfrac{N}{19}\right) + \tfrac{3332}{171} N - \tfrac{38}{9}(-1)^{\log_2\left(\frac{N}{19}\right)} + 38, \\ C_N^{\times}(19 \times 2^p) &= \tfrac{19}{3} 2^{2+p} p - \tfrac{19}{9} 2^{1+p} + \tfrac{38}{9}(-1)^p + 114 = \\ &= \tfrac{4}{3} N \log_2\left(\tfrac{N}{19}\right) - \tfrac{38}{171} N + \tfrac{38}{9}(-1)^{\log_2\left(\frac{N}{19}\right)} + 114. \end{aligned} \tag{23}$$

4. 2/8 Split-Radix Algorithm

In case of $a = 4$, the algorithm becomes 2/8 split-radix algorithm.

$$X[8k + l] = \sum_{n=0}^{N/4-1} W_N^{nl} \Big(\sum_{n=0}^{3} x'[n + i\frac{N}{8}] W_8^{il}\Big) W_{N/8}^{nk}. \tag{24}$$

The total numbers of real multiplications and real additions needed by the algorithm are

$$\begin{aligned} C_N^{+} &= C_{N/2}^{+} + 4C_{N/8}^{+} + \tfrac{11}{2} N - C_t^{+} \\ C_N^{\times} &= C_{N/2}^{\times} + 4C_{N/8}^{\times} + \tfrac{5}{2} N - C_t^{\times}. \end{aligned} \tag{25}$$

Below the number of arithmetic operations in logarithmic form are presented

$$\begin{aligned} C_N^{+} &= \tfrac{11}{4} pq2^p - \tfrac{55}{16} q2^p - \tfrac{1}{8} 2^p \left(C_t^{+} - (2C_q^{+} + C_{2q}^{+} C_{4q}^{+})\right) + \tfrac{1}{4} C_t^{+} + \\ &+ (-1)^p 2^{p/2} \quad [7(12C_q^{+} - 2(C_{2q}^{+} + C_{4q}^{+} + C_t^{+}) + 55q)\cos(p \arctan(\sqrt{7})) \\ &+ \sqrt{7}(4C_q^{+} - 2(11C_{2q}^{+} - 5C_{4q}^{+} - C_t^{+}) - 99q)\sin(p \arctan(\sqrt{7}))] \\ C_N^{\times} &= \tfrac{5}{4} pq2^p - \tfrac{25}{16} q2^p - \tfrac{1}{8} 2^p \left(C_t^{\times} - (2C_q^{\times} + C_{2q}^{\times} C_{4q}^{\times})\right) + \tfrac{1}{4} C_t^{\times} + \\ &+ (-1)^p 2^{p/2} \quad [7(2(C_t^{\times} - 6C_q^{\times} + C_{2q}^{\times} + C_{4q}^{\times}) - 25q)\cos(p \arctan(\sqrt{7})) \\ &+ \sqrt{7}(C_t^{\times} + 4C_q^{\times} - 22C_{2q}^{\times} + 5C_{4q}^{\times} - 45q)\sin(p \arctan(\sqrt{7}))]. \end{aligned} \tag{26}$$

where by C_q,C_{2q} and C_{4q} denote number of arithmetic operations required for computations of q, $2q$ and $4q$ length DFTs, respectively. For computing $2q$ and $4q$ length DFT, we can use

methods described in the previous section, which allows to rewrite (26)

$$\begin{aligned} C_N^+ &= \tfrac{11}{4}pq2^p - \tfrac{15}{16}q2^p - \tfrac{1}{8}2^p\left(C_t^+ - 8C_q^+\right) + \tfrac{1}{4}C_t^+ + \\ &+(-1)^p 2^{p/2} \quad [7(15q - 2C_t^+)\cos(p\arctan(\sqrt{7})) \\ &+\sqrt{7}(2C_t^+ - 27q)\sin(p\arctan(\sqrt{7}))] \\ C_N^\times &= \tfrac{5}{4}pq2^p - \tfrac{25}{16}q2^p - \tfrac{1}{8}2^p\left(C_t^\times - 8C_q^\times\right) + \tfrac{1}{4}C_t^\times + \\ &+(-1)^p 2^{p/2} \quad [7(25q - 2C_t^\times)\cos(p\arctan(\sqrt{7})) \\ &+\sqrt{7}(2C_t^\times - 45q)\sin(p\arctan(\sqrt{7}))]. \end{aligned} \tag{27}$$

4.1 Complexity of 2/8 Split-Radix Algorithm for $q < 20$

In case of $q = 15$, we have $C_{15}^+ = 168$, $C_{15}^\times = 30$ and from [7] $C_t^+ = 180$, $C_t^\times = 60$ and therefore

$$\begin{aligned} C_N^+ &= \tfrac{11}{4}pq2^p - \tfrac{15}{16}q2^p - \tfrac{1}{8}2^p\left(C_t^+ - 8C_q^+\right) + \tfrac{1}{4}C_t^+ + \\ &+(-1)^p 2^{p/2} \quad [7(15q - 2C_t^+)\cos(p\arctan(\sqrt{7})) \\ &+\sqrt{7}(2C_t^+ - 27q)\sin(p\arctan(\sqrt{7}))] \\ C_N^\times(15\times 2^p) &= \tfrac{5}{4}p15\times 2^p - \tfrac{17}{16}15\times 2^p - \tfrac{15}{16}(-1)^p 2^{p/2}() + 45 \\ &+(-1)^p 2^{p/2} \quad [7(25q - 2C_t^\times)\cos(p\arctan(\sqrt{7})) \\ &+\sqrt{7}(2C_t^\times - 45q)\sin(p\arctan(\sqrt{7}))]. \end{aligned} \tag{28}$$

5. Comparison

The number of additions and multiplications required for computing DFT for various lengths are presented in Table 1 (2/4 split-radix algorithm). As an example a range from 256 to 2048 is chosen. Using only conventional algorithm for 2^p we have only compute DFT of $256, 512, 1024, 2048$ sizes. If size is not equal to these values we need to pad the input data up to next 2^p. Using $q \times 2^p$ algorithm allows to cover the range $256 - 1024$ with 27 new points. This approach allows to significantly reduce the number of arithmetic operations. To find out the q for which the algorithm becomes the most efficient in terms of the number of arithmetic operations, first of all we cut the values of q for which $C_{N_1} > C_{N_2}$, but $N_1 < N_2$, where $C_N = C_N^+ + C_N^\times$. It is easy to see that only for $1, 3, 5, 9, 13, 15$ condition presented above is true. For getting more accurate results we compare the value of $E_N = \frac{C_N}{N}$. Finally we get

$$\begin{aligned} E_N(9\times 2^p) < E_N(9\times 3^p) < E_N(9\times 15^p) < E_N(1\times 2^p) < \\ < E_N(5\times 2^p) < E_N(15\times 2^p). \end{aligned}$$

Table 1: Number of arithmetic operations required by 2/4 split-radix algorithm for the DFT length $256 - 1024$

N	q	p	Add.	Mul.	Count
256	1	8	5380	1284	6664
272	17	4	6832	1720	8552
288	9	5	6036	1196	7232
304	19	4	9200	1672	10872
320	5	6	6736	1880	8616
352	11	5	9468	2204	11672
360	45	3	7812	1140	8952
384	3	7	8028	2192	10220
416	13	5	10852	2372	13224
448	7	6	10992	2760	13752
480	15	5	10956	2112	13068
512	1	9	12292	3076	15368
544	17	5	15092	4052	19144
576	9	6	13584	3136	16720
608	19	5	19996	4028	24024
640	5	7	15172	4580	19752
704	11	6	20784	5288	26072
720	45	4	17424	3000	20424
768	3	8	18096	5396	23492
832	13	6	23888	5784	29672
896	7	7	24364	6668	31032
960	15	6	24432	5460	29892
1024	1	10	27652	7172	34824
1088	17	6	33040	9464	42504
1152	9	7	30228	7724	37952
1216	19	6	43184	9576	52760
1280	5	8	33744	10840	44584
1408	11	7	45308	12380	57688
1440	45	5	38628	7620	46248
1536	3	9	40284	12816	53100
1664	13	7	52196	13700	65896
1792	7	8	53488	15688	69176
1920	15	7	53964	13344	67308
2048	1	11	61444	16388	77832

In Table 2 are presented the comparison results for the various length DFTs (2/8 split-radix algorithm).

Table 2: Number of arithmetic operations required by 2/4 split-radix algorithm for the DFT length $256 - 2048$

N	q	p	Add.	Mul.	Count
256	1	8	5380	1284	6664
360	45	3	8048	1360	9408
480	15	5	11256	2520	13776
512	1	9	12292	3076	15368
720	45	4	18076	3620	21696
960	15	6	25212	6180	31392
1024	1	10	27652	7172	34824
1440	45	5	39696	8640	48336
1920	15	7	55824	14880	70704
2048	1	11	61444	16388	77832

6. Conclusion

In case of looking for computationally efficient algorithm in terms of number of multiplications in general case we need to choose 2/8 split-radix FFT algorithm, because coefficient of $N \log_2$ is smaller. Efficiency of algorithms in terms of total number of arithmetic operations is discussed below.
The total number of arithmetic operations required by 2/4 split-radix algorithm can be computed using (12) and is presented below

$$C_N(2/4) = 4pq2^p - 2^p(6q - C_q^+) + 8q. \tag{29}$$

The total number of arithmetic operations required for computation 2/8 split-radix algorithm can be retrieved from (28)

$$\begin{aligned} C_N(2/8) = {} & 4pq2^p - 2^p(\tfrac{5}{2}q - (\tfrac{1}{8}C_t - C_q)) + 8q + \\ & +2(-1)^p 2^{p/2} \times [7(20q - C_t)\cos(\alpha) + \\ & +\sqrt{7}(C_t - 36q)\sin(\alpha).] \end{aligned} \tag{30}$$

For getting a computationally efficient algorithm, we need to calculate $C_N(2/4) - C_N(2/8)$ using (29) and (30). For simplicity, only the coefficients for 2^p and $q \times 2^p$ are included

$$\begin{gathered} C_N(2/4) - C_N(2/8) = (\tfrac{7}{2}q - \tfrac{1}{8}C_t)2^p < 0 \\ C_t > 28q. \end{gathered}$$

In other words we can say that if C_t is greater than $28q$, 2/4 split-radix algorithm is more efficient compared to $C_N(2/8)$.

References

[1] E. O. Brigham, *The Fast Fourier Applications*, Englewood Cliffs, NJ, Prentice-Hall, 1988.

[2] J. K. Ersoy, *Fourier-Related Transforms. Fast Algorithms and Applications*, Englewood Cliffs, NJ, Prentice-Hall,1997.

[3] J. W. Cooley and J. W. Tukey, "An algorithm for the machine computation of the complex Fourier series," *Math. Computation*, pp. 297–301, 1965.
[4] R. Yavne, "An economical method for calculating the discrete Fourier transform," *in Proc. AFIPS*, vol. 33, pp. 115–125, 1968.
[5] M. Frigo and S. G. Johnson, "A modified split-radix FFT with fewer arithmetic operations", *Ieee Trans. Signal Processing*, vol. 55, pp. 111-119, 2207.
[6] G. Bi and Y. Q. Chen, "Fast DFT algorithm for length $N = q \times 2^p$," *IEEE Trans. Circuits and Systems II*, vol. 45, no. 6, pp. 685 - 690, 1998.
[7] G. Bi, G. Li and X. Li, "A unified expression for split-radix DFT algorithms," pp. 323 - 326, 2010.
[8] P. Duhamel, "Implementation of split-radix FFT algorithms for complex, real, and real-symmetric data", *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 34, pp. 285 - 295, April, 1986.
[9] H. V. Sorensen, M. T. Heideman and C. S. Burrus, "On computing the split-radix FFT," IEEE Trans. Acoust., Speech, Signal Process., vol. 34,pp. 152 - 156, Feb. 1986.
[10] S. Bouguezel, M. Omair and M. N. S. Swamy, "A new radix-2/8 FFT algorithm for length-$q \times 2^m$ DFTs," *IEEE Trans. Circuits and Systems I*, vol. 51, no. 1, pp. 1723-1732, 2004.
[11] S. Bouguezel, M. Omair and M. N. S. Swamy, "A general class of split-radix FFT algorithms for the computation of the DFT of length-2^m," *IEEE Trans. Signal Processing*, vol. 55, no. 8, pp. 4127 - 4138, 2007.
[12] Online: [Available] http://maxima.sourceforge.net
[13] I. Selesnick and S. Burrus, "Programs for Prime Length FFTs," http://cnx.org/content/m18137/1.5/.
[14] R. Barseghyan, "Complexity of the composite length FFT algorithms", *Proceedings of International Conference CSIT 2015*, Yerevan, Armenia, 2015.

Submitted 22.08.2017, accepted 06.12.2017.

$q \times 2^p$ - երկարության ՖԱՁ-բարդության մասին

Ռ. Բարսեղյան

Ամփոփում

Աշխատանքում ստացված է լոգարիթմական բանաձևը, որը կամայական $q \times 2^p$-երկարության վեկտորների համար թույլ է տալիս հաշվարկել Ֆուրյեի դիսկրետ ձևափոխության (ՖԱՁ) համար անհրաժեշտ գործողությունների ճշգրիտ քանակը, որտեղ q-ն կենտ թիվ է:

О сложности алгоритмов БПФ для длины $q \times 2^p$

Р. Барсегян

Аннотация

В этой статье выведена логарифмическая формула, которая позволяет вычислить точное количество необходимых операций для вычисления дискретного преобразования Фурье (DFT) для векторов длины $q \times 2^p$, где q - нечетное целое число.