Mathematical Problems of Computer Science 33, 24–34, 2010.

New Approach to FFT Algorithms

Rafayel Barseghyan and Hakob Sarukhanyan

Institute for Informatics and Automation Problems of NAS RA

Abstract

In this paper we present a new, efficient modification of split-radix algorithm for computing a power of two discrete Fourier transforms. The developed algorithm allows to 40% real arithmetic operations reduction in comparison with previous best results for 16-point discrete Fourier transform.

1. Introduction

Applications of linear transforms, such as Fourier, Hadamard, Cosine and Sine transforms in signal and image processing are numerous [1]. Cooley and Tukey published their historic paper on the computation of the Fourier transform in 1965. Overnight, in universities and laboratories around the world, scientists and engineers began developing computer programs and electronic circuits to implement the FFT. The FFT is a brilliant technique for computing the discrete Fourier (DFT) transform quickly. By recognizing that the Fourier transform of a sequence can be derived from the Fourier transforms of two half length sequences more economically than if the whole sequence is transformed directly and by carrying this concept through to its logical conclusion of evaluating only the direct transform of sequences of two terms, Cooley and Tukey showed that the FFT required only $O(N \log N)$ operations while the direct form took $O(N^2)$ operations.Any improvement in FFT algorithms appears to rely on reducing the exact number or cost of these operations rather than their asymptotic functional form [2]. For many years, the time to perform an FFT was dominated by real-number arithmetic, and so considerable effort was devoted towards proving and achieving lower bounds on the exact count of arithmetic operations (real additions and multiplications),called "flops" (floating-point operations), required for a DFT of a given size [3],[4]. Although the performance of FFTs on recent computer hardware is determined by many factors besides pure arithmetic counts, there still remains an intriguing unsolved mathematical question: what is the smallest number of flops required to compute a DFT of a given size N?

2. 2^p–point FFT

2..1 Conventional Case

Let $x = \{x_0, x_1, \ldots, x_{N-1}\}^T$ be a complex valued column-vector of length N ($N = 2^p$). The forward and inverse 1D DFT of this vector are defined as

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

where $W_N^n = \exp(-j\frac{2\pi}{N}n) = \cos(\frac{2\pi}{N}n) - j\sin(\frac{2\pi}{N}n)$, $j = \sqrt{-1}$.

Represent the forward transform as follows (here and later the coefficient 1/N is omitted)

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

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

Introduce the notations:

$$Y_0[n] = \sum_{k=0}^{N/2-1} x[2k]W_{N/2}^{nk}, \qquad Y_1[n] = \sum_{k=0}^{N/2-1} x[2k+1]W_{N/2}^{nk}, \quad n = \overline{0, N/2-1}. \tag{3}$$

Note that $Y_0[n]$ and $Y_1[n]$ are $N/2$–point forward DFT.

Hence, the equation (2) can be represented as follows

$$X[n] = Y_0[n] + W_N^n Y_1[n], \qquad X[n+N/2] = Y_0[n] - W_N^n Y_1[n], n = \overline{0, N/2-1}. \tag{4}$$

It is easy to show that

$$W_N^0 = 1, \quad W_N^{N/8} = \frac{\sqrt{2}}{2}(1-j), \quad W_N^{N/4} = -j, \quad W_N^{3N/8} = -\frac{\sqrt{2}}{2}(1+j).$$

Therefore, the realization of $W_N^n Y_1[n]$, for all n needs only $(N-4)$-real addition and $(2N-12)$-real multiplication operations. Storing this results we can calculate the necessary operations for N–point DFT given in equation (4), i.e. we obtain

$$C_N^+ = 3N - 4 + 2C_{N/2}^+, \qquad C_N^\times = 2N - 12 + 2C_{N/2}^\times, \tag{5}$$

where C_N^+ and $C_N^\times$ denotes the number of additions and multiplications of N–point DFT, respectively.

Finally, from relations (5) we can obtain

$$C_N^{+} = 3N\log_2 N - 3N + 4, \qquad C_N^{\times} = 2N\log_2 N - 7N + 12, \quad N \geq 8. \tag{6}$$

Note that $C_2^{+} = 4, C_2^{\times} = 0$, $C_4^{+} = 16, C_4^{\times} = 0$. In the next table some numerical results are given

Table 1

N	Add	Mul	Total
2	4	0	4
4	16	0	16
8	52	4	56
16	148	28	176
32	388	108	496
64	964	332	1296
128	2308	908	3216
256	5380	2316	7696
512	12292	5644	17936
1024	27652	13324	40976
2048	61444	30732	92176
4096	135172	69644	204816
8192	294916	155660	450576
16384	638980	344076	983056
32768	1376260	753676	2129936

3. Modified FFT

3..1 Conventional Case

Let $x = \{x_0, x_1, \ldots, x_{N-1}\}^T$ be a complex valued column-vector of length N ($N = 2^p$). The DFT of this vector can be represented as (the coefficient $1/N$ is omitted)

$$X[n] = \sum_{k=0}^{\frac{N}{2}-1} x[2k]W_{\frac{N}{2}}^{nk} + W_N^n \sum_{k=0}^{\frac{N}{4}-1} x[4k+1]W_{\frac{N}{4}}^{nk} + W_N^{3n} \sum_{k=0}^{\frac{N}{4}-1} x[4k+3]W_{\frac{N}{4}}^{nk}. \tag{7}$$

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

It is not difficult to show that with assumption $x[-1] = x[N-1]$ we have

$$W_N^{3n} \sum_{k=0}^{\frac{N}{4}-1} x[4k+3]W_{\frac{N}{4}}^{nk} = W_N^{-n} \sum_{k=0}^{\frac{N}{4}-1} x[4k-1]W_{\frac{N}{4}}^{nk}.$$

Therefore the equation (7) we can rewrite as following

$$X[n] = \sum_{k=0}^{\frac{N}{2}-1} x[4k]W_{\frac{N}{2}}^{nk} + W_N^n \sum_{k=0}^{\frac{N}{4}-1} x[4k+1]W_{\frac{N}{4}}^{nk} + W_N^{-n} \sum_{k=0}^{\frac{N}{4}-1} x[4k-1]W_{\frac{N}{4}}^{nk}, \tag{8}$$

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

Introduce the following notations:

$$\begin{aligned} A_N^n &= W_N^n Y_1[n] + W_N^{-n} Y_2[n], \\ S_N^n &= W_N^n Y_1[n] - W_N^{-n} Y_2[n], && n = \overline{0, N/4-1}; \\ Y_0[n] &= \sum_{k=0}^{N/2-1} x[4k] W_{N/2}^{nk}, && n = \overline{0, N/2-1}; \\ Y_1[n] &= \sum_{k=0}^{N/4-1} x[4k+1] W_{N/4}^{nk}, && , \\ Y_2[n] &= \sum_{k=0}^{N/4-1} x[4k-1] W_{N/4}^{nk}, && n = \overline{0, N/4-1}. \end{aligned} \tag{9}$$

Hence, $N = 2^p$–point DFT can be computed by the following formulae

$$\begin{aligned} X[n] &= Y_0[n] + A_N^n, \\ X[n + \tfrac{N}{4}] &= Y_0[n + \tfrac{N}{4}] - jS_N^n, \\ X[n + \tfrac{2N}{4}] &= Y_0[n] - A_N^n, \\ X[n + \tfrac{3N}{4}] &= Y_0[n + \tfrac{N}{4}] + jS_N^n, \quad n = \overline{0, N/4-1}. \end{aligned} \tag{10}$$

3..2 Complexity Evaluation

Now we calculate the necessary operations for DFT presented in (10). At first using the properties of exponential function W we have

$$W_N^0 = 1, \quad W_N^{N/8} = \frac{\sqrt{2}}{2}(1-j).$$

Therefore, the realization of A_N^n requires $\frac{3}{2}N - 4$ and $2N - 12$ addition and multiplication operations, respectively. The realization of S_N^n requires only $N/2$ additions.

Thus, the necessary operations for realization N–point DFT presented in (10) can be obtained from the following formulae

$$\begin{aligned} C_N^+ &= 4N - 4 + C_{N/2}^+ + 2C_{N/4}^+, \\ C_N^\times &= 2N - 12 + C_{N/2}^\times + 2C_{N/4}^\times, \quad N \geq 8. \end{aligned} \tag{11}$$

Using the theory of difference equations [6] we obtain

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

In Table 2 some numerical results are given

Table 2

N	Add	Mul	Total
2	4	0	4
4	16	0	16
8	52	4	56
16	144	24	168
32	372	84	456
64	912	248	1160
128	2164	660	2824
256	5008	1656	6664
512	11380	3988	15368
1024	25488	9336	34824
2048	56436	21396	77832
4096	123792	48248	172040
8192	269428	107412	376840
16384	582544	236664	819208
32768	1252468	517012	1769480

4. New FFT algorithm with fewer flops

4..1 Efficient Implementation of FFT

We will perform DFT by two steps. At first we introduce some notations:

$$\begin{aligned} T_{N,n} &= [1 - j\tan\tfrac{2\pi}{N}n]. \\ C[N,n] &= \cos(\tfrac{2\pi}{N}[n \bmod N/4]). \end{aligned} \tag{13}$$

where $Y_1[n]$, $Y_2[n]$ are given in (9). Note that

$$W_N^n = T_{N,n}\cos\frac{2\pi}{N}n.$$

Using this notations now we represent $N = 2^p$–point DFT from (10) by the following two steps

Step 1: $n = 0, 1, \ldots, N/4 - 1$.

$$\begin{aligned} X[n] &= Y_0[n] + (W_N^n C[N/4,n]Y_1[n] + W_N^{-n}C[N/4,n]Y_2[n]), \\ X[n+\tfrac{N}{4}] &= Y_0[n+\tfrac{N}{4}] - j(W_N^n C[N/4,n]Y_1[n] - W_N^{-n}C[N/4,n]Y_2[n]), \\ X[n+\tfrac{2N}{4}] &= Y_0[n] - (W_N^n C[N/4,n]Y_1[n] + W_N^{-n}C[N/4,n]Y_2[n]), \\ X[n+\tfrac{3N}{4}] &= Y_0[n+\tfrac{N}{4}] + j(W_N^n C[N/4,n]Y_1[n] - W_N^{-n}C[N/4,n]Y_2[n]); \end{aligned} \tag{14}$$

Step 2: $n = 0, 1, \ldots, N/16 - 1$.

$$\begin{aligned} Y_1[n] &= Y_{10}[n]/\cos\tfrac{2\pi}{N/4}n + (T_{N/4,n}Y_{11}[n] + T^*_{N/4,n}Y_{12}[n]), \\ Y_1[n+\tfrac{N}{16}] &= Y_{10}[n+\tfrac{N}{16}]/\cos\tfrac{2\pi}{N/4}n - j(T_{N/4,n}Y_{11}[n] - T^*_{N/4,n}Y_{12}[n]), \\ Y_1[n+\tfrac{2N}{16}] &= Y_{10}[n]\cos\tfrac{2\pi}{N/4}n - (T_{N/4,n}Y_{11}[n] + T^*_{N/4,n}Y_{12}[n]), \\ Y_1[n+\tfrac{3N}{16}] &= Y_{10}[n+\tfrac{N}{16}]/\cos\tfrac{2\pi}{N/4}n + j(T_{N/4,n}Y_{11}[n] - T^*_{N/4,n}Y_{12}[n]), \end{aligned} \tag{15}$$

$$\begin{aligned} Y_2[n] &= Y_{20}[n]/\cos\tfrac{2\pi}{N/4}n + (T_{N/4,n}Y_{21}[n] + T^*_{N/4,n}Y_{22}[n]),\\ Y_2[n+\tfrac{N}{16}] &= Y_{20}[n+\tfrac{N}{16}]/\cos\tfrac{2\pi}{N/4}n - j(T_{N/4,n}Y_{21}[n] - T^*_{N/4,n}Y_{22}[n]),\\ Y_2[n+\tfrac{2N}{16}] &= Y_{20}[n]\cos\tfrac{2\pi}{N/4}n - (T_{N/4,n}Y_{21}[n] + T^*_{N/4,n}Y_{22}[n]),\\ Y_2[n+\tfrac{3N}{16}] &= Y_{20}[n+\tfrac{N}{16}]/\cos\tfrac{2\pi}{N/4}n + j(T_{N/4,n}Y_{21}[n] - T^*_{N/4,n}Y_{22}[n]), \end{aligned} \tag{16}$$

4..2 Complexity evaluation

Now we calculate the necessary operations for DFT presented in (14)-(16). At first using the properties of cosine and exponential functions we obtain

$$W_N^n \cos\frac{2\pi}{N/4}n = 1, \text{ if } n = 0. \tag{17}$$

where $n = 0, 1, \ldots, N/4 - 1$.

For $n = 0, 1, \ldots, N/16 - 1$ we have

$$\begin{aligned} \cos\tfrac{2\pi}{N/4}n &= \begin{cases} 1, & \text{if } n = 0,\\ \frac{\sqrt{2}}{2}, & \text{if } n = N/32, \end{cases}\\ T_{N/4,n} &= \begin{cases} 1, & \text{if } n = 0,\\ 1-j, & \text{if } n = N/32. \end{cases} \end{aligned} \tag{18}$$

Therefore, without taking the operations for $Y_0[n]$, $Y_1[n]$, and $Y_2[n]$, we can calculate the necessary real operations for computing the terms $X[n]$, $X[n+\frac{N}{4}]$, $X[n+\frac{2N}{4}]$, and $X[n+\frac{3N}{4}]$ from equation (14) for all $n = \overline{0, N/4-1}$(see Table below).

Table 3

Expression	Add	Mul
$X[n]$	$2N-4$	$2N-8$
$X[n+\frac{N}{4}]$	N	0
$X[n+\frac{2N}{4}]$	$\frac{1}{2}N$	0
$X[n+\frac{3N}{4}]$	$\frac{1}{2}N$	0

Now we can calculate the number of real operations for computing all component $X[n]$ ($n = \overline{0, N-1}$, $N \geq 16$)

$$\begin{aligned} C_X^+ &= 4N - 4 + C_{Y_0}^+ + C_{Y_1}^+ + C_{Y_2}^+,\\ C_X^\times &= 2N - 8 + C_{Y_0}^\times + C_{Y_1}^\times + C_{Y_2}^\times, \end{aligned} \tag{19}$$

where $C_{Y_0}^+$ and $C_{Y_0}^\times$ are the complexity of $N/2-$point DFT, and $C_{Y_1}^+$, $C_{Y_1}^\times$ and $C_{Y_2}^+$, $C_{Y_2}^\times$ are the complexity of transforms given in (15) and (16), respectively.

Now we define the necessary real operations for the terms

$$T_{N/4,n}Y_{11}[n] \pm T^*_{N/4,n}Y_{12}[n],\ T_{N/4,n}Y_{21}[n] \pm T^*_{N/4,n}Y_{22}[n]$$

without taking the operations for terms $Y_{11}[n]$, $Y_{12}[n]$, $Y_{21}[n]$, and $Y_{22}[n]$ (see (15) and (16).

At first we have

$$T_{N/4,0} = 1, \quad T_{N/4,N/32} = 1 - j.$$

Hence, we obtain

Table 4

Expression	Add	Mul
$T_{N/4,n}Y_{11}[n]+T^*_{N/4,n}Y_{12}[n]$	$\frac{3}{8}N-4$	$\frac{1}{4}N-8$
$T_{N/4,n}Y_{11}[n]-T^*_{N/4,n}Y_{12}[n]$	$\frac{1}{8}N$	0
$T_{N/4,n}Y_{21}[n]+T^*_{N/4,n}Y_{22}[n]$	$\frac{3}{8}N-4$	$\frac{1}{4}N-8$
$T_{N/4,n}Y_{21}[n]-T^*_{N/4,n}Y_{22}[n]$	$\frac{1}{8}N$	0

Now using the results of Table 4 without taking the operations for $Y_{i,j}[n],\ i,j=0,1,2$ (see (15) and (16) we can define the operations for realization of $Y_1[n], n=\overline{0,N/4-1}$ (see Table below).

Table 5

Expression	Add	Mul
$Y_1[n]$	$\frac{1}{2}N-4$	$\frac{3}{8}N-10$
$Y_1[n+\frac{N}{16}]$	$\frac{1}{4}N$	$\frac{1}{8}N-2$
$Y_1[n+\frac{2N}{16}]$	$\frac{1}{8}N$	0
$Y_1[n+\frac{3N}{16}]$	$\frac{1}{8}N$	0

Note that for $Y_2[n]$ the number of required operations is the same as for $Y_1[n]$. Now using the results of Table 5 we can calculate the number of real operations for computing all components of $Y_1[n]$ and $Y_2[n]$ ($n=\overline{0,N/4-1}$, $N\geq 32$)

$$\begin{aligned} C^+_{Y_1} &= N-4+C^+_{Y_{10}}+C^+_{Y_{11}}+C^+_{Y_{12}}, \\ C^\times_{Y_1} &= \tfrac{1}{2}N-12+C^\times_{Y_{10}}+C^\times_{Y_{11}}+C^\times_{Y_{12}}. \end{aligned} \tag{20}$$

It is not difficult to show that

$$\begin{aligned} &C^+_X = C^+_N,\ C^\times_X = C^\times_N, \\ &C^+_{Y_0} = C^+_{N/2},\ C^\times_{Y_0} = C^\times_{N/2}, \\ &C^+_{Y_{10}} = C^+_{Y_{20}} = C^+_{N/8}, \\ &C^\times_{Y_{10}} = C^\times_{Y_{20}} = C^\times_{N/8}, \\ &C^+_{Y_{11}} = C^+_{Y_{21}} = C^+_{Y_{12}} = C^+_{Y_{22}} = C^+_{N/16}, \\ &C^\times_{Y_{11}} = C^\times_{Y_{21}} = C^\times_{Y_{12}} = C^\times_{Y_{22}} = C^\times_{N/16}. \end{aligned} \tag{21}$$

Finally using the equations (19),(20) and the identities (21) we obtain the complexity of $N-$point DFT as

$$\begin{aligned} C^+_N &= 6N-12+C^+_{N/2}+2C^+_{N/8}+4C^+_{N/16}, \\ C^\times_N &= 3N-32+C^\times_{N/2}+2C^\times_{N/8}+4C^\times_{N/16}. \end{aligned} \tag{22}$$

Optimization by hand for $N=16$ has allowed us to save 32-additions and 16-multiplications in comparison with algorithm 3 (see, section 3). Using these results and relations (22) we can obtain

$$\begin{aligned} C^+_N &= \tfrac{8}{3}N\log_2 N-\tfrac{8}{3}N-\tfrac{8}{9}\alpha^+_N\sqrt{N}-\tfrac{34}{9}(-1)^{\log_2 N}+2, \\ C^\times_N &= \tfrac{4}{3}N\log_2 N-\tfrac{41}{9}N-\alpha^\times_N\sqrt{N}-\tfrac{16}{9}(-1)^{\log_2 N}+\tfrac{16}{3}. \end{aligned} \tag{23}$$

Values of α^+_N and $\alpha^\times_N$ are defined in Table 6

Table 6

$\log_2 N (mod 4)$	α_N^{+}	$\alpha_N^{\times}$
0	4	2
1	$\sqrt{2}$	$\frac{\sqrt{2}}{3}$
2	-4	-2
3	$-\sqrt{2}$	$-\frac{\sqrt{2}}{3}$

In Table 7 some numerical results are given.

Table 7

N	Add	Mul	Total
16	112	8	120
32	332	72	404
64	872	240	1112
128	2044	624	2668
256	4648	1536	6184
512	10780	3808	14588
1024	24488	9056	33544
2048	54236	20736	74972
4096	118952	46752	165704
8192	260188	104640	364826
16384	564904	231456	796360
32768	1216348	506176	1722524

5. Complex multiplication with 3 real multiplications

5..1 Modified complex multiplication

Below a method is presentedwhich allows us to do complex multiplication with 3 real multiplications and 5 real additions. Multiplication of two complex numbers (a+jb) and (c+jd) means

$$(a + jb)(c + jd) = (ac - bd) + j(ad + bc) \tag{24}$$

We can represent the real part of (24) us as the following

$$(a - b)d + a(c - d). \tag{25}$$

And the complex part

$$(a - b)d + b(c + d). \tag{26}$$

For realization of (25) we have 2 real multiplications and 3 real additions. (26) requires 1 real multiplication and 2 real multiplication given that $(a-b)d$ is already computed. Finally we get 3 real multiplication and 5 real additions.

Assuming that our second complex number $(c + jd)$ has form $(\sin\phi + j\cos\phi)$.Now assuming that $(\sin\phi + \cos\phi)$ and $(\sin\phi - \cos\phi)$ are pre-computed this scheme can take only 3-real multiplications and 3-real additions (More about FFT algorithms that use scheme 3-multiplications and 3-additions see [7],[8]).

5..2 Complexity evaluation

Using methods presented in section 4.2 for first step we get these recurrence relations

$$C_X^{+} = \tfrac{11}{2}N - 10 + C_{Y_0}^{+} + C_{Y_1}^{+} + C_{Y_2}^{+},$$
$$C_X^{\times} = \tfrac{3}{2}N - 6 + C_{Y_0}^{\times} + C_{Y_1}^{\times} + C_{Y_2}^{\times}, \tag{27}$$

For evaluating 2-step of algorithm we use standard multiplication scheme with 4 real additions and 2 real multiplications. Finally we get

$$C_N^{+} = \tfrac{15}{2}N - 18 + C_{N/2}^{+} + 2C_{N/8}^{+} + 4C_{N/16}^{+},$$
$$C_N^{\times} = \tfrac{5}{2}N - 30 + C_{N/2}^{\times} + 2C_{N/8}^{\times} + 4C_{N/16}^{\times}. \tag{28}$$

$$C_N^{+} = \tfrac{10}{3}N\log_2 N - \tfrac{29}{6}N - \tfrac{1}{18}\alpha_N^{+}\sqrt{N} - \tfrac{47}{9}(-1)^{\log_2 N} + 3.$$
$$C_N^{\times} = \tfrac{10}{9}N\log_2 N - \tfrac{23}{6}N - \tfrac{1}{54}\alpha_N^{\times}\sqrt{N} - \tfrac{35}{27}(-1)^{\log_2 N} + 5. \tag{29}$$

Values of α_N^{+} and $\alpha_N^{\times}$ are defined in Table 8

Table 8

$\log_2 N (mod 4)$	α_N^{+}	$\alpha_N^{\times}$
0	98	74
1	$11\sqrt{2}$	$23\sqrt{2}$
2	-98	-74
3	$-11\sqrt{2}$	$-23\sqrt{2}$

Some numerical results are shown in Table 9

Table 9

N	Add	Mul
32	382	58
64	1012	196
128	2386	518
256	5500	1276
512	12874	3150
1024	29356	7500
2048	65242	17214
4096	143692	38828
8192	315322	86878
16384	686092	192236
32768	1480186	420638

6. FFT for Real Input Vector

For computation DFT of real input we can use algorithms of complex FFTs. Algorithms which compute real-FFT via complex-FFT are simple and based on property of hermitian-symmetry

$$X[n] = X^{*}[N-n], \tag{30}$$

where $X^*[n]$ is a conjugate of $X[n]$.
The complexity of real-FFT is

$$C_N^+(R) = \frac{1}{2}C_N^+ - (N-2),$$
$$C_N^\times(R) = \frac{1}{2}C_N^\times. \tag{31}$$

Using 31 we can get

- for the algorithm in algorithm 2(presented in Section 2)

$$C_N^+(R) = \frac{3}{2}N\log_2 N - \frac{5}{2}N + 4,$$
$$C_N^\times(R) = N\log_2 N - \frac{7}{2}N + 6. \tag{32}$$

- for the algorithm in algorithm 3

$$C_N^+(R) = \frac{4}{3}N\log_2 N - \frac{25}{9}N - \frac{1}{9}(-1)^{\log_2 N} + 4.$$
$$C_N^\times(R) = \frac{2}{3}N\log_2 N - \frac{19}{9}N + \frac{1}{9}(-1)^{\log_2 N} + 3. \tag{33}$$

- for the algorithm in algorithm 4

$$C_N^+(R) = \frac{4}{3}N\log_2 N - \frac{11}{3}N - \frac{4}{9}\alpha_N^+\sqrt{N} - \frac{17}{9}(-1)^{\log_2 N} + 4,$$
$$C_N^\times(R) = \frac{2}{3}N\log_2 N - \frac{41}{18}N - \frac{1}{2}\alpha_N^\times\sqrt{N} - \frac{8}{9}(-1)^{\log_2 N} + \frac{8}{3}. \tag{34}$$

where values of α_N^+ and $\alpha_N^\times$ are defined in Table 6.

We can use methods in [6] to obtain fast algorithms and their complexities (such as 32, 33, 34) of discrete sine,cosine and Hartley transforms from presented algorithm.

7. Comparison results

In 2007 Steven Johnson and Matteo Frigo presented [3] new FFT algorithm which has fewer arithmetic operations than all known FFT algorithms. Graphical presentation of comparison results for algorithm 4,Johnson-Frigo algorithm and algorithm 3.

Algorithm 4 requires fewer arithmetic operations up to $N < 2^{13}$

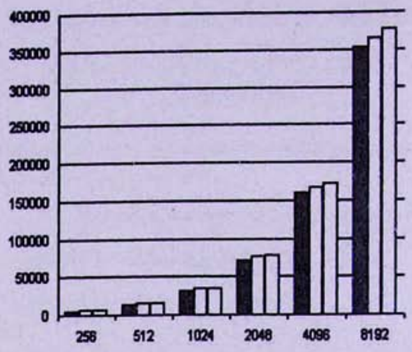

Figure 1. Flop counts of algorithm 4, Johnson-Frigo algorithm, algorithm 3.

References

[1] R. Blahut, Fast Algorithms for Digital Signal Processing. Addison-Wesley, 1985.

[2] P. Duhamel and M. Vetterli, Fast Fourier transforms: a tutorial review and a state of the art, Signal Processing - Apr., vol. 19, pp. 259 299, 1990.

[3] M. Frigo and S. G. Johnson, "A modified split-radix FFT with fewer arithmetic operations", IEEE Trans. Signal Proccessing, vol. 55, pp. 111-119, 2007.

[4] H. Sarukhanyan, S. Agaian, "Conventional, Integer to Integer and Quantized Fast Fourier Transforms ", CSIT, pp. 204-207, 2007.

[5] I. G. Petrovski, Lectures on the theory of ordinary differential equations , (in russian), 1984.

[6] M. Vetterli, H. J. Nussbaumer, "Simple FFT and DCT algorithms with reduced number of operations", *Signal Processing* , vol. 6, Nr. 4, pp. 267-278, 1984.

[7] R.Barseghyan, "FFT with lifting transforms", *9-th annual conference, RAU* , (submitted), 2009.

[8] R.Barseghyan, "Split-radix FFT algorithm with lifting steps", *Vestnik RAU, Series: Physics-mathematics and natural science*, pp. 42-50, 2009.

Նոր մոտեցում ՖԱՁ ալգորիթմների

Ռ. Բարսեղյան և Հ. Սարուխանյան

Ամփոփում

Աշխատանքում մշակված է կտրտված հիմքով Ֆուրյեի արագ ձևափոխության նոր, արդյունավետ ձևափոխված ալգորիթմը: Ձևափոխված ալգորիթմի հիման վրա օպտիմալացվել է 16-չափանի վեկտորի ձևափոխությունը, որի հետևանքով պահանջվող իրական թվաբանական գործողությունների քանակը նվազել է 40% -ով: