Tikrit Journal of Engineering Sciences (2023) 30 (4): 127-133
DOI: http://doi.org/10.25130/tjes.30.4.13

IRAQI Academic Scientific Journals

College of Engineering

ISSN: 1813-162X (Print); 2312-7589 (Online)

Tikrit Journal of Engineering Sciences

available online at: http://www.tj-es.com

TJES Tikrit Journal of Engineering Sciences

New Algorithm for Real-Valued Fourier Transform

Sukaina K. Salih, Mounir T. Hamood *
Department of Electrical Engineering, College of Engineering, Tikrit University, Tikrit, Iraq.

Keywords:

Real-Valued; Fast Fourier Transform; RDFT; Decimation-In-Time; Fast Algorithm.

ARTICLE INFO

Article history:

Received	23 Apr.	2023
Received in revised form	17 May	2023
Accepted	03 Sep.	2023
Final Proofreading	23 Nov.	2023
Available online	07 Dec.	2023

Citation: Salih SK, Hamood MT. **New Algorithm for Real-valued Fourier Transform**. *Tikrit Journal of Engineering Sciences* 2023; **30**(4): 127-133.
http://doi.org/10.25130/tjes.30.4.13

***Corresponding author:**

Mounir T. Hamood

Department of Electrical Engineering, College of Engineering, Tikrit University, Tikrit, Iraq.

Abstract: This paper presents a direct algorithm for fast real discrete Fourier transform (RDFT) computing, using the discrete Fourier transform (DFT) conjugate symmetric property to reduce redundancies. In RDFT, all the input and output signals were real, which differed from complex DFT. Therefore, the structure of the proposed algorithm showed only real-data operations. The developed algorithm showed the desired properties, such as in-place computation, regularity, simplicity, and arithmetic operations reduction. The RFFT performance was compared with other related transforms, such as the fast Hartley transform (FHT) for the computation in the radix-2 algorithm. It was found that FHT showed the best performance in terms of arithmetic complexity.

خوارزمية جديدة لتحويل فورير ذو القيمة الحقيقية

سكينة خزعل صالح، منير طه حمود
قسم الهندسة الكهربائية/ كلية الهندسة/ جامعة تكريت / تكريت- العراق.

الخلاصة

في هذا البحث، تم تقديم خوارزمية للحساب السريع لتحويل فورير المنفصل الحقيقي بطريقة مباشرة. حيث تم استغلال الخاصية المتماثلة المقترنة لتقليل التكرارات. في RFFT تكون جميع إشارات الادخال والاخراج حقيقة والتي تختلف عن تحويل فورير المنفصل المعقد، لذلك فان الرسم البياني المتدفق للخوارزمية المقترحة يمتلك عمليات للبيانات الحقيقية فقط. يوضح هيكل الفراشة للخوارزمية المقترحة الخصائص المرغوبة مثل الحساب الموضعي والانتظام في الهيكل مع تقليل عدد التعقيدات الحسابية. تمت مقارنة RFFT بأداء تحويلات أخرى مماثلة مثل تحويل هارتلي السريع FHT للحساب في الجذر-٢ ووجد أنهُ يتمتع بأداء أفضل من ناحية التعقيد الحسابي.

الكلمات الدالة: القيمة الحقيقية، تحويل فورير السريع، RDFT، التقسيم الزمني، الخوارزمية السريعة.

1.INTRODUCTION

The discrete Fourier transform (DFT) is an important tool in electrical engineering, especially in digital signal processing, high-performance computation, and communication [1]. The DFT is used to analyze nonperiodic signals to produce a frequency spectrum of those signals. It converts a sequence in the time domain to its frequency domain components, whereas the inverse discrete Fourier transform (IDFT) retrieves the original signal. The Fast Fourier Transform (FFT) is an effective algorithm for calculating DFT, and it represents a fast approach that can achieve the same result as the DFT with fewer operations [2]. In FFT algorithms, there are two approaches: Decimation-in-Time (DIT) and Decimation-in-Frequency (DIF). In DIT, the input data $x(n)$ of length N can be separated into two sequences, $x_1(n)$ and $x_2(n)$. They correspond to the input $x(n)$'s even-indexed and odd-indexed samples, respectively. While in DIF, the length N is divided into two parts: the first part contains the first half ($N/2$) data, and the second part contains the other half of the sequence $x(n)$. These algorithms can be employed for computing the DFT for lengths with integer powers of two [3]. The fast RDFT algorithms development reflects their importance as they have been used in many applications, such as radar signals processing [4], spectrum sensing [5], artificial neural networks [6], nonlinear matched filtering [7], and analyzing the signal of biomedical of data [8]. The RDFT has all the properties of the DFT. In addition, it deals with real signals, as most natural applications utilize real signals [9, 10]. The conventional real-valued FFT used two different ways to calculate the real data. The first method converts real data into complex data by adding zeros to the imaginary part; the second method depends on transferring the real sequence to a complex sequence by dividing the length N by two. In this case, the even and odd parts of the input sequence represent the real and imaginary parts sequentially. More calculations are required to find the real input DFT from the complex sequence DFT. Despite the simplicity and ease of the DFT method in real-data calculations, it requires many operations. However, there are many algorithms to compute the real signal DFT [11-14]. When the DFT deals with a real signal, the appearance of conjugate symmetry properties in the output is noticeable, which means there is a redundancy in the calculations. Utilizing the conjugate symmetry of the DFT, the operations can be reduced by almost half [15, 16]. It is found that $N/2 - 1$ are redundant calculations using the DFT. The RDFT utilization reduces the time required to calculate the arithmetic operations and the memory storage by half. Later, a brief literature is introduced to compute RFFT algorithms [17-21]. The main objective of this paper is to develop a new algorithm for computing the real-valued of the fast Fourier transform and compare the arithmetic complexity of RFFT with the fast Hartley transform. Section 2 lists previous works on RDFT. Section 3 reviews the real discrete transforms. The proposed RFFT on decimation in time algorithm is derived in Section 4. Section 5 presents the complex arithmetic of the developed algorithm. Finally, the conclusion is given in Section 6.

2.PREVIOUS STUDIES

Various RFFT computation algorithms have been demonstrated in the past. Most of these studies were interested in the RFFT implementation in-place and pipelined. This section presents some of the literature on RFFT. Lao and Parhi [22] presented a real-valued algorithm (RFFT) using the canonic property to design RFFT, which was suitable. The canonic RFFT approach required the least number of butterfly operations, and it increased the regularity of architectures, which decreased the hardware complexity. Yin et al. [23] found a new algorithm for a pair of parallel pipelined radix-2 RFFT implemented depending on real datapaths, which differs from hybrid datapaths and fully utilized hardware resources. Garrido et al. [15] designed a pipelined architecture to compute the real signal of a fast Fourier transform where the input signal deals serially. The proposed architecture was mentioned as a real-valued serial commutator (RSC) FFT. This approach provided 50% hardware consumption and gave high exploitation of butterflies. Kumar et al. [24] discussed an efficient algorithm for FFT called quick Fourier transform (QFT) by taking advantage of the symmetric properties of the DFT. The QFT algorithm had an interesting structure related to the discrete Cosine transform (DCT) and discrete Sine transform (DST). It calculated the DFT using DCT and DST separately and worked effectively for the DFT on real data. This algorithm represented an indirect method to compute the DFT for real data. Park and Jeon [25] modified the real serial commutator (mRSC) architecture for FRFT, reducing processing delay and device complexity. The mRSC designs could not enable real-valued IFFT (RIFFT) in their original structure. Instead, they only accepted real-valued inputs and processed real and imaginary components separately in each stage. The suggested architecture showed the least hardware complexity and implementation cost compared with pipelined FFT architectures for serial real-valued. Majorkowska-Mech and Cariow [26] presented a real-valued algorithm for computing FFT of small length *N* from (3 to 9) and explained the data flow graph of each length. Their strategy of the algorithms was expressed in matrix-vector notation, where the matrices of RDFT were factorized, and the factors were sparse matrices. The number of arithmetic operations was decreased by this factoring. Eleftheriadis and Karakonstantis [27] proposed a new architecture for FFT real-valued applications that would save energy using the DFT properties for odd conjugate symmetric. The architecture approach represented decimate in time, where the input was in time with bit reverse order. When processing two inputs simultaneously, it reduces the memory cells number. Therefore, it was possible to double the throughput with only a slight increase in area overhead.

3.REAL DISCRETE TRANSFORMS

3.1.Real Discrete Fourier Transform (RDFT)

Discrete Fourier Transform (DFT) for length *N* of sequence $x(n)$ is defined as [24]:

$$X(k) = \sum_{n=0}^{N-1} x(n)\, W_N^{nk} \quad k=0,1,2,\dots,N-1 \qquad (1)$$

Where $W_N = e^{j2\pi/N}$

$X(k)$ and $x(n)$ in DFT represent complex signals (both have phase and amplitude). The DFT is based on sine waves [28]. In the case of RDFT, all the inputs of sequence *x(n)* are real, making the output $X(k)$ conjugate symmetric.

$$X(N-K) = X^*(K) \quad 1 \le k \le N/2 - 1 \qquad (2)$$

Where *X* (*0*) and *X*(*N*/2) are real output signals, and other outputs are conjugate symmetric. By taking advantage of this feature, all duplicates can be removed. As a result, an *N*-point RFFT only needs to compute *N*/2+1 output. The forward RDFT can be defined as [26]:

$$X(k) = \sum_{n=0}^{N-1} x(n) \cos\left(\frac{2\pi nk}{N} + \theta(k)\right) \qquad (3)$$

Where

$$\theta(k) = \begin{cases} 0, & 0 \le k \le N/2 \\ \frac{\pi}{2} & \frac{N}{2} < k \le N-1 \end{cases} \qquad (4)$$

The inverse RDFT is defined as:

$$x(n) = \frac{2}{N}\sum_{k=0}^{N-1} X(k)\, v(n) \cos\left(\frac{2\pi nk}{N} + \theta(k)\right) \qquad (5)$$

Where

$$v(n) = \begin{cases} \frac{1}{2} & n = 0, N/2 \\ 1, & \text{otherwise} \end{cases} \qquad (6)$$

3.2.Discrete Hartley Transform (DHT)

The discrete Hartley transform (HT) represents an orthogonal transformation nearly identical to the Fourier transform in many ways. The DHT is based on the DFT, and they have a close relationship. The DHT has features similar to those of the DFT. FHT is one type of real transform popularly utilized with real signals. It depends on the same basic functions as the DFT. However, the DHT employs just real sinusoids and cosines rather than complex exponentials. The formula for the DHT pair is expressed as [29]:

$$X(k) = \sum_{n=0}^{N-1} x(n)\, cas\left(\frac{2\pi nk}{N}\right) \quad k=0,1,2,\dots,N-1 \qquad (7)$$

Where $cas\left(\frac{2\pi nk}{N}\right) = \sin\left(\frac{2\pi nk}{N}\right) + \cos\left(\frac{2\pi nk}{N}\right)$.

Observe that the only difference between the HT and the DFT is the absence of the"-j," meaning that the DHT corresponds to directly subtracting the imaginary term from the real term of the DFT of a real-valued sequence. From Eqs. (3) and (7), the RDFT is simpler to compute the real signals DFT than the DHT. It has been shown that the RFFT and the HT are closely related. Also, each RFFT algorithm has an equivalent algorithm for Hartley and vice versa. The Hartley transform has the same forward and inverse, which makes it useful on processors with limited memory [14].

4.THE PROPOSED ALGORITHM

In RDFT, it is assumed that the length *N* is 2^t, i.e., t is an integer. From Eq. (3), the length is separated into even and odd sequences, so it is possible to write.

$$X_R(k) = X_e(k) + X_o(k) \quad \textbf{(8)}$$

$$X_e(k) = \sum_{n=0}^{N/2-1} x(2n)\cos\left(\frac{2\pi nk}{N/2} + \theta(k)\right) \quad \textbf{(9)}$$

Eq. (9) represents the RDFT, so

$$X_e(k) = X_{2n}(k) \quad \textbf{(10)}$$

$$X_o(k) = \sum_{n=0}^{N/2-1} x(2n+1)\cos\left(\frac{2\pi k(n+\frac{1}{2})}{N/2} + \theta(k)\right) \quad \textbf{(11)}$$

$$X_o(k) = \sum_{n=0}^{N/2-1} x(2n+1)\cos\left\{\left(\frac{2\pi nk}{N/2} + \theta(k)\right) + \frac{2\pi k}{N}\right\} \quad \textbf{(12)}$$

Using the following trigonometric identities:

$$\cos(A+B) = \cos A\,\cos B - \sin A\,\sin B \quad \textbf{(13)}$$

It can be obtained that:

$$X_O = \sum_{n=0}^{N/2-1} x(2n+1)\cos\frac{\pi k}{N}\cos\left(\frac{2\pi nk}{N/2} + \theta(k)\right) - \sum_{n=0}^{N/2-1} x(2n+1)\sin\frac{\pi k}{N}\sin\left(\frac{2\pi nk}{N/2} + \theta(k)\right) \quad \textbf{(14)}$$

From the RDFT definition, the first summation can be written as:

$$X_o(k) = X_{2n+1}(k)\cos\frac{\pi k}{N} - \sin\frac{\pi k}{N}\sum_{n=0}^{N/2-1} x(2n+1)\sin\left(\frac{2\pi nk}{N/2} + \theta(k)\right) \quad \textbf{(15)}$$

To simplify the second summation, using the definition of $\theta(k)$ given in Eq. (4):

$$\theta(k) = \begin{cases} 0 & 0 \le k \le N/2 \\ \frac{\pi}{2} & N/2 < k \le N-1 \end{cases} \quad \textbf{(16)}$$

$$\sin\left(\frac{2\pi nk}{N/2} + \theta(k)\right) = \begin{cases} \sin\frac{2\pi nk}{N/2} & 0 \le k \le N/2 \\ \cos\frac{2\pi nk}{N/2} & N/2 < k \le N-1 \end{cases} \quad \textbf{(17)}$$

And

$$\cos\left(\frac{2\pi nk}{N/2} + \theta(k)\right) = \begin{cases} \cos\frac{2\pi nk}{N/2} & 0 \le k \le N/2 \\ -\sin\frac{2\pi nk}{N/2} & N/2 < k \le N-1 \end{cases} \quad \textbf{(18)}$$

From these relationships, the second summation of Eq. (14) can be converted to:

$$\sum_{n=0}^{N/2-1} x(2n+1)\sin\left(\frac{2\pi nk}{N/2} + \theta(k)\right) = \sum_{n=0}^{N/2-1} x(-(2n+1)\cos\left(\frac{2\pi nk}{N/2} + \theta(k)\right) \quad \textbf{(19)}$$

Using the DFT periodicity property

$$-X(k) = X(N-k) \quad \textbf{(20)}$$

yields to

$$X_o(k) = X_{2n+1}(k)\cos\frac{\pi k}{N} - X_{2n+1}(N/2-k)\sin\frac{\pi k}{N} \quad \textbf{(21)}$$

Therefore, the first equation of the proposed algorithm is:

$$X_R(k) = X_{2n}(k) + X_{2n+1}(k)\cos\frac{\pi k}{N} - X_{2n+1}(N/2-k)\sin\frac{\pi k}{N} \quad \textbf{(22)}$$

In the same manner, the derivation of the DIT-RFFT algorithm found for other decompositions are:

$$X_R\left(\frac{N}{2}-k\right) = X_{2n}(k) - X_{2n+1}(k)\cos\frac{\pi k}{N} + X_{2n+1}(N/2-k)\sin\frac{\pi k}{N} \quad \textbf{(23)}$$

$$X_R\left(\frac{N}{2}+k\right) = -X_{2n}(N/2-k) + X_{2n+1}(k)\sin\frac{\pi k}{N} + X_{2n+1}(N/2-k)\cos\frac{\pi k}{N} \quad \textbf{(24)}$$

$$X_R(N-k) = X_{2n}(N/2-k) + X_{2n+1}(k)\sin\frac{\pi k}{N} + X_{2n+1}(N/2-k)\cos\frac{\pi k}{N} \quad \textbf{(25)}$$

The resulting butterfly of the proposed algorithm is shown in Fig. 1.

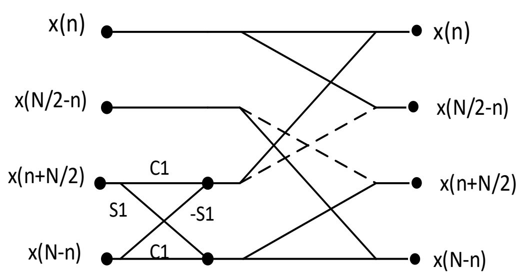

Fig. 1 Real-Valued Fast Fourier Transform (RFFT) Butterfly.

The strategy followed by the proposed algorithm is shown in Fig. 2.

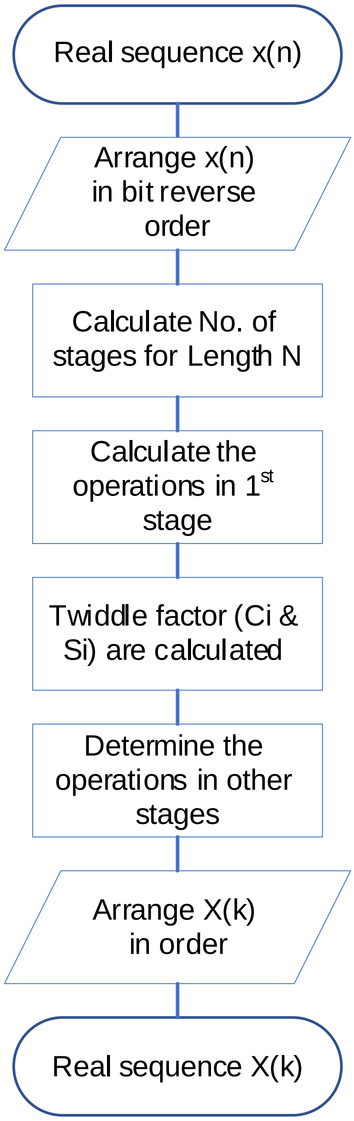

Fig. 2 The Flowchart of the Proposed Algorithm (RFFT).

5.ARITHMETIC COMPLEXITY

In this section, the proposed algorithm is analyzed by calculating the number of operations. The real multiplication R_M and real additions R_A were computed from a single flow graph (SFG) in Fig.3. Generally, the developed algorithm required ($\log_2 N$) stages of butterfly computation. In the first stage, there were N additions, followed by $N/2$ additions in the second stage, and no multiplication in these stages. Each butterfly in all stages had two reductions, except in the first stage, where the reductions in the last stage occurred at points $N/4$ and $3N/4$. The complexity can be calculated as:

$$R_M(N) = N(\log_2 N - 2) - (N - 4) \quad \textbf{(26)}$$

$$R_A(N) = \frac{3N}{2}(log_2 N - 2) + \frac{3N}{2} - (N - 4) \quad \textbf{(27)}$$

Simplification of Eq. (26) and Eq. (27) yields

$$R_M(N) = N\,log_2 N - 3N + 4 \quad \textbf{(28)}$$

$$R_A(N) = \frac{3N}{2} log_2 N - \frac{5N}{2} + 4 \quad \textbf{(29)}$$

Table 1 compares the proposed algorithm RFFT with FHT in terms of computational complexity. The number of multiplications $H_M(N)$ and the number of additions $H_A(N)$ for FHT [30] are given as:

$$H_M(N) = N\,log_2 N - 3N + 4 \quad \textbf{(30)}$$

$$H_A(N) = (3N\,log_2 N - 3N + 4)/2 \quad \textbf{(31)}$$

The RFFT algorithm complexity was calculated using Eqs. (28) and (29). This comparison shows that FHT requires (N-2) additions more than the RFFT algorithm. The proposed RFFT algorithm uses the standard two multiplications and four additions scheme, and it is based on the single-butterfly structure that requires a simple hardware or software structure.

Table 1 Comparison between RFFT and FHT.

Transform length(N)	Radix-2 RFFT			Radix-2 FHT		
	Mults.	**Adds.**	**Mult+Add**	**Mults.**	**Adds**	**Mult+Add**
8	4	20	24	4	26	30
16	20	60	80	20	74	94
32	68	164	232	68	194	262
64	196	420	616	196	482	678
128	516	1028	1544	516	1154	1670
256	1284	2436	3720	1284	2690	3974
512	3076	5636	8712	3076	6146	9222
1024	7172	12804	19976	7172	13826	20998

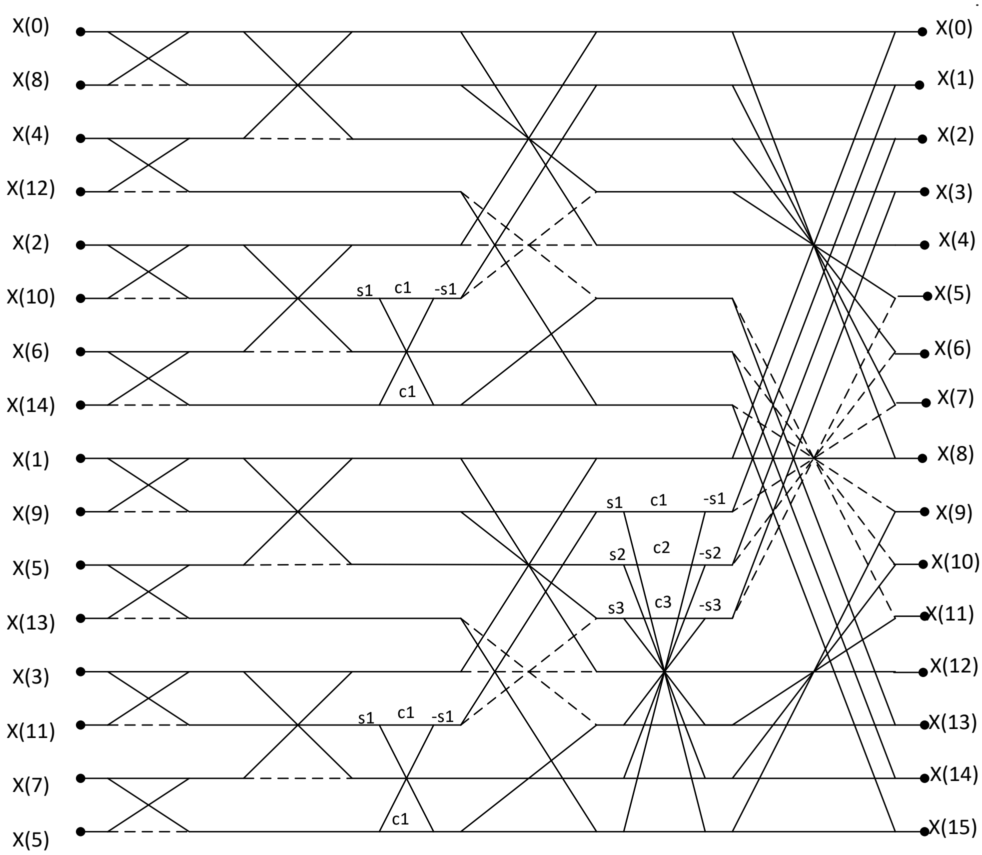

Fig. 3 Radix-2 RFFT Signal Flow Diagram when N=16, where $si = \sin\frac{2\pi i}{N}$ and $ci = \cos\frac{2\pi i}{N}$, Respectively.

6.CONCLUSION

An efficient RFFT decimation-in-time algorithm was proposed. By exploiting the property of conjugate symmetry for the real signal of the DFT, all duplicates and symmetries were eliminated. The arithmetic complexity of the suggested algorithm was calculated and carefully checked. It was compared to the number of operations using the FHT approach. This comparison showed that the proposed RFFT radix-2 algorithm outperformed the number of arithmetic operations and structural complexities compared to FHT radix-2 algorithms.

REFERENCES

[1] Behrendt M, de Angelis M, Comerford L, Zhang Y, Beer M. **Projecting Interval Uncertainty through the Discrete Fourier Transform: An Application to Time Signals with Poor Precision**. *Mechanical Systems and Signal Processing* 2022; **172**:108920, (1-36).

[2] Özhan O. (2022). **Fast Fourier Transform**. In *Basic Transforms for Electrical Engineering* (pp. 465-494). Cham: Springer International Publishing.

[3] Takahashi D. (2019). **Fast Fourier Transform**. *Fast Fourier Transform Algorithms for Parallel Computers*, 5-13.

[4] Roienko O, Lukin V, Oliinyk V, Djurović I, Simeunović M. **An Overview of the Adaptive Robust DFT and it's Applications**. *Technological Innovation in Engineering Research* 2022; **4**:68-89.

[5] Khayyeri M, Mohammadi K. **Cooperative Wideband Spectrum Sensing in Cognitive Radio Based on Sparse Real-Valued Fast Fourier Transform**. *IET Communications* 2020;**14**(8):1340-1348.

[6] Feng Q, Liu Y, Lai Y-K, Yang J, Li K. **FOF: Learning Fourier Occupancy Field for Monocular Real-Time Human Reconstruction**. *Advances in Neural Information Processing Systems* 2022;**35**:7397-7409.

[7] Moryakova O, Johansson H. **Frequency-Domain Implementations of Variable Digital FIR Filters Using the Overlap-Save Technique**. *2023 24th International Conference on Digital Signal Processing (DSP)*: IEEE; Rhodes (Rodos), Greece, 2023. pp. 1-5.

[8] Tang SN, Jan FC, Cheng HW, Lin CK, Wu GZ. **Multimode Memory-Based FFT Processor for Wireless Display FD-OCT Medical Systems**. *IEEE Transactions on Circuits and Systems I: Regular Papers* 2014;**61**(12):3394-3406.

[9] Mohanty BK, Meher PK. **Area–Delay–Energy Efficient Vlsi Architecture**

for Scalable in-Place Computation of FFT on Real Data. *IEEE Transactions on Circuits and Systems I: Regular Papers* 2018; **66**(3):1042-1050.

[10] Ma ZG, Yin XB, Yu F. **A Novel Memory-Based FFT Architecture for Real-Valued Signals Based on a Radix-2 Decimation-in-Frequency Algorithm**. *IEEE Transactions on Circuits and Systems II: Express Briefs* 2015; **62**(9):876-880.

[11] Hamood MT. **Improved FHT Algorithms for Fast Computation of the Discrete Hartley Transform**. *Tikrit Journal of Engineering Sciences* 2013; **20**(1):62-69.

[12] Bouguezel S, Ahmad MO, Swamy M. **Binary Discrete Cosine and Hartley Transforms**. *IEEE Transactions on Circuits and Systems I: Regular Papers* 2012; **60**(4):989-1002.

[13] Britanak V, Yip PC, Rao KR. **Discrete Cosine and Sine Transforms: General Properties, Fast Algorithms and Integer Approximations**: Elsevier; Oxford, UK, 2010.

[14] Sorensen HV, Jones D, Heideman M, Burrus C. **Real-Valued Fast Fourier Transform Algorithms**. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 1987; **35**(6):849-863.

[15] Garrido M, Unnikrishnan NK, Parhi KK. **A Serial Commutator Fast Fourier Transform Architecture for Real-Valued Signals**. *IEEE Transactions on Circuits and Systems II: Express Briefs* 2017; **65**(11):1693-1697.

[16] Chinnapalanichamy A, Parhi KK. **Serial and Interleaved Architectures for Computing Real FFT**. *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*: IEEE; South Brisbane, QLD, Australia, 2015. pp. 1066-1070.

[17] Fang H, Zhang B, Yu F, Zhao B, Ma Z. **A Pipelined Algorithm and Area-Efficient Architecture for Serial Real-Valued FFT**. *IEEE Transactions on Circuits and Systems II: Express Briefs* 2022; **69**(11):4533-4537.

[18] Sanjeet S, Sahoo BD, Parhi KK. **Comparison of Real-Valued FFT Architectures for Low-Throughput Applications Using FPGA**. *2021 IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*: IEEE; Lansing, MI, USA, 2021. pp. 112-115.

[19] Garrido M. **A Survey on Pipelined FFT Hardware Architectures**. *Journal of Signal Processing Systems* 2022; **94**(11):1345-1364.

[20] Salehi SA, Amirfattahi R, Parhi KK. **Pipelined Architectures for Real-Valued FFT and Hermitian-Symmetric IFFT with Real Datapaths**. *IEEE Transactions on Circuits and Systems II: Express Briefs* 2013; **60**(8):507-511.

[21] Ayinala M, Lao Y, Parhi KK. **An in-Place FFT Architecture for Real-Valued Signals.** *IEEE Transactions on Circuits and Systems II: Express Briefs* 2013; **60**(10):652-656.

[22] Lao Y, Parhi KK. **Data-Canonic Real FFT Flow-Graphs for Composite Lengths**, *2016 IEEE International Workshop on Signal Processing Systems (SiPS)*, Dallas, TX, USA, 2016, pp. 189-194.

[23] Yin XB, Yu F, Ma ZG. **Resource-Efficient Pipelined Architectures for Radix-2 Real-Valued FFT with Real Datapaths**. *IEEE Transactions on Circuits and Systems II: Express Briefs* 2016; **63**(8):803-807.

[24] Kumar GG, Sahoo SK, Meher PK. **50 Years of FFT Algorithms and Applications**. *Circuits, Systems, and Signal Processing* 2019; **38**:5665-5698.

[25] Park S, Jeon D. **A Modified Serial Commutator Architecture for Real-Valued Fast Fourier Transform**. *In 2020 IEEE Workshop on Signal Processing Systems (SiPS)*; Coimbra, Portugal, 2020, (pp. 1-6). IEEE.

[26] Majorkowska-Mech D, Cariow A. **Some FFT Algorithms for Small-Length Real-Valued Sequences**. *Applied Sciences* 2022; **12**(9):4700, (1-21).

[27] Eleftheriadis C, Karakonstantis G. **Energy-Efficient Fast Fourier Transform for Real-Valued Applications**. *IEEE Transactions on Circuits and Systems II: Express Briefs* 2022; **69**(5):2458-2462.

[28] Al-Ubaidi SM. **Design and Implementation of Sequency-Ordered Fast Walsh-Hadamard Transform (FWHT) in WCDMA**. *Tikrit Journal of Engineering Sciences* 2014; **21**(3):26-31.

[29] Gautam D. **Resourceful Fast Discrete Hartley Transform to Replace Discrete Fourier Transform with Implementation of DHT Algorithm for VLSI Architecture**. *Turkish Journal of Computer and Mathematics Education* 2021; **12**(10) :5290-5298.

[30] Jones KJ. **The Regularized Fast Hartley Transform: Low-Complexity Parallel Computation of the FHT in One and Multiple Dimensions**. Springer Nature: 2021; pp. 35-50.