Research on algorithm and implementation of Fourier transform in the case of $N < M$

Yang Gao1*, JiaLi Zhu1, YunFei Li1, Jian Liu1

1No.724 Research Institute of CSSC, Chang Qing Road 30, Jiang Ning District, Nanjing City, Jiangsu Province, 211106, People’s Republic of China

*gaoyang_3006@163.com

Abstract. In this paper, Fourier transform (FT) is studied under a special situation in which the transform length $N$ is less than the number of input data $M$. First, the corresponding Fourier transform formula is deduced, and then an implementation example is given as well as performance simulation and implementation analysis. The results show that the mixed radix FT method has less resource consumption and less processing latency compared with fast Fourier transform (FFT) and discrete Fourier transform (DFT) method, respectively. The proposed $N < M$ FT method increases the flexibility of traditional Fourier transform algorithm.

1. Introduction

Fourier transform (FT) is a common algorithm in the field of signal processing and is widely used in image processing [1], digital communication [2], signal spectrum analysis [3], and array signal processing [4], etc. The traditional implementation of Fourier transform is generally based on fast Fourier transform (FFT) method which can reduce the complexity of implementation. FFT method is a radix-2 algorithm with an inconvenience that the transform length should be the integer power of 2. If the number of input data does not meet this requirement, zero padding is enforced. This determines that the transform length of FFT must be larger than the number of input data. However, in some special applications, such as digital signal spectrum analysis, increasing the transform length by zero padding causes a denser output spectrum and increases the number of spectral lines passing the threshold, which finally increases the pressure of signal detection and fusion processing.

In this paper, Fourier transform algorithm and its implementation under a special requirement that the transform length is less than the number of input data is studied. The rest of the paper is organized as follows. In Section 1, the corresponding Fourier transform formula is derived. Several mixed radix FT algorithm as well as an implementation example are described in Section 2. Simulation and performance analysis are carried out in Section 3. At last, a summary is given.

2. The principle of Fourier transform: from the perspective of transform length

Let the length of signal $x(n)$ be $M$ and perform $N$-point Fourier transform on it. When $N = M$, the discrete Fourier transformation (DFT) formula is

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

When $N > M$, the above formula is rewritten as

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{nk}, k = 0, 1, ..., N - 1$$
\[
X(k) = \sum_{n=0}^{N-1} x(n)W_N^{nk}, \quad k = 0, 1, \ldots, N-1
\]

Where when \(n = 0, 1, \ldots, M-1\), \(\bar{x}(n) = x(n)\); when \(n = M, M+1, \ldots, N-1\), \(\bar{x}(n) = 0\), \(N\) is usually taken as an integer power of 2, and the formula can be implemented using FFT method.

When \(N < M\), the Fourier transform formula is changed to:

\[
X(k) = \sum_{n=0}^{M-1} x(n)W_N^{nk}, \quad k = 0, 1, \ldots, N-1
\]

\[
= \sum_{n=0}^{N-1} x(n)W_N^{nk} + \sum_{n=N}^{M-1} x(n)W_N^{nk}
\]

\[
= \sum_{n=0}^{N-1} x(n)W_N^{nk} + \sum_{n=0}^{M-1-N} x(n+N)W_N^{nk} + \sum_{n=M-N}^{N-1} 0W_N^{nk}
\]

Where when \(n = 0, 1, \ldots, M-N-1\), \(\bar{x}(n) = x(n) + x(N+n)\); when \(n = N, N+1, \ldots, M-1\), \(\bar{x}(n) = x(n)\). In this situation, the \(N\)-point Fourier transform of the \(M\)-point input signal \(x(n)\) becomes the \(N\)-point Fourier transform of \(\bar{x}(n)\), and the spectrum output is not distorted. In this case, the transform length \(N\) is not necessarily the integer power of 2, in hence, the FFT method cannot be used.

### 3. Mixed Radix Fourier Transformation and Implementation

When the transform length \(N\) is not an integer power of 2, several fast transform algorithms can be utilized to realize DFT operation, such as Cool-Turkey algorithm [5], Good-Thomas algorithm [6], and Winograd Fourier Transform Algorithm (WFTA) algorithm [7-8]. If the number of input data \(M\) is 80, and the transform length \(N\) is 75, a 75-point FT is applied. \(N\) is decomposed as follows: \(75 = (5 \times 5) \times 3\), in which the \((25 \times 3)\) point Fourier transform is based on the Good-Thomas Algorithm, and the \((5 \times 5)\) point Fourier transform is based on the Cooley-Turkey algorithm, and the 5-point and 3-point Fourier transform are implemented based on the 5-point and 3-point WFTA algorithm, respectively.

#### 3.1 Cooley-Turkey Algorithm

The core of Cooley-Turkey Algorithm is to map one-dimensional Fourier transformation into two-dimensional transformation, i.e., the input and output index of \(N\)-point DFT can be decomposed as follows:

\[n = N_1 \cdot n_1 + n_2 (0 \leq n_1 \leq N_1, 0 \leq n_2 \leq N_2)\]

\[k = k_1 + N_1 \cdot k_2 (0 \leq k_1 \leq N_1, 0 \leq k_2 \leq N_2)\]

Where \(N = N_1N_2\). Hence, the original rotation factor of DFT is rewritten as

\[W_N^{nk} = W_N^{N_1n_1k_1+N_1n_2k_2}W_N^{n_1k_1}W_N^{N_1n_2k_2}\]

\[= W_N^{n_1k_1}W_N^{N_1n_2k_2}\]

Hence,

\[X(k) = X(k_1, k_2) = \sum_{n_1=0}^{N_1-1} W_N^{n_1k_1} \left( \sum_{n_2=0}^{N_2-1} x(n_1, n_2)W_N^{n_2k_2} \right)\]

where one dimensional DFT \(X(k)\) is transformed into two-dimensional DFT \(X(k_1, k_2)\). Two points need to be paid attention to in the implementation of the above formula: first, the order of input and output
data need to be adjusted; Second, after the first $N_1$-point FT, extra phase rotation operation is required before the $N_2$-point FT.

In this $(5 \times 5)$-point FT example, let
\[
\begin{align*}
n &= T_1 \cdot n_1 + T_2 \cdot n_2 \\
k &= T_3 \cdot k_1 + T_4 \cdot k_2
\end{align*}
\]
where $n = 0, 1, \ldots, N-1; k = 0, 1, \ldots, N-1; n_1, k_1 = 0, 1, \ldots, N-1; n_2, k_2 = 0, 1, \ldots, N-1$;
The result is $T_1 = N_2, T_2 = 1, T_3 = 1, T_4 = N_1$. Hence, the index mapping follows $n = 5n_1 + n_2$, and $k = k_1 + 5k_2$.

### 3.2 Good-Thomas Algorithm

When the translation length $N$ can be decomposed into two coprime factors $N_1$ and $N_2$, the calculation can be further simplified by mapping the input and output subscripts $n$ and $k$ as:
\[
\begin{align*}
n &= T_1 \cdot n_1 + T_2 \cdot n_2 (0 \leq n_1 \leq N_1, 0 \leq n_2 \leq N_2) \\
k &= T_3 \cdot k_1 + T_4 \cdot k_2 (0 \leq k_1 \leq N_1, 0 \leq k_2 \leq N_2)
\end{align*}
\]
Where $T_1, T_2 = \text{mod}(0, N)$, $T_3, T_4 = \text{mod}(0, N)$, $T_1, T_3 = \text{mod}(N_2, N)$, $T_2, T_4 = \text{mod}(N_1, N)$.
As in the example, $T_2 = N_1 = 25$, $T_1 = N_2 = 3$, then
\[
\begin{align*}
T_4 &= 0 + N_1 \cdot k_1 \\
T_3 &= 1 + N_2 \cdot k_2
\end{align*}
\]
The minimum $T_4$ obtained by above formula is the required value.
\[
\begin{align*}
T_4 &= 0 + N_2 \cdot k_2 \\
T_3 &= 1 + N_1 \cdot k_1
\end{align*}
\]
The minimum $T_3$ obtained by above formula is the required value.
According to the above calculation, $T_1 = 3, T_2 = 25, T_3 = 51, T_4 = 25$;
In this $(25 \times 5)$-point FT example, the mapping follows $n = 3n_1 + 25n_2$, $k = 51k_1 + 25k_2$.

### 3.3 WFTA

WFTA is an algorithm for small length inputs and can be used to calculate $N = 2, 3, 4, 5, 7, 8, 9, 16$, etc. Its core idea is to reduce the number of addition, subtraction, multiplication, and division of DFT by declining the dimension of transformation matrix. The transformation matrix $W$ of DFT is decomposed into:

\[
W = ODI
\]
Where $D$ is a diagonal matrix with real or imaginary elements; elements of $O$ and $I$ matrix are 0, ±1 or some small integers.

Taking 5-point WFTA as an example, $W$ can be decomposed into:
\[
W = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & -1 \\
1 & 1 & -1 & 1 & 0 \\
1 & 1 & -1 & -1 & 0 \\
1 & 1 & 1 & -1 & 1
\end{bmatrix} \begin{bmatrix}
b_0 \\
b_1 \\
b_2 \\
b_3 \\
b_4
\end{bmatrix} = \begin{bmatrix}
1 & 1 & 1 & 1 \\
0 & 1 & 1 & 1 \\
0 & 1 & -1 & -1 \\
0 & 0 & -1 & 1 \\
0 & 0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
b_0 \\
b_1 \\
b_2 \\
b_3 \\
b_4
\end{bmatrix}
\]
where, $b_0 = 1$, $b_1 = (\cos \theta + \cos 2\theta)/2 - 1$, $b_2 = (\cos \theta - \cos 2\theta)/2$, $b_3 = -j \sin \theta$, $b_4 = -j (\sin \theta + \sin 2\theta)$, $b_5 = j (\sin \theta - \sin 2\theta)$, $\theta = 2\pi/5$.

### 3.4 Implementation and resource consumption

The implementation structure of the 75-point FT example is depicted in Figure 1. As shown in this figure, the implementation requires 30 WFTA_5 modules and 25 WFTA_3 modules in total. For $M$-point digital signal, in order to realize $N$-point DFT, $M \cdot N$ complex multiplication and complex
addition are needed. The resource consumption of $N$-point FFT only operates $N \cdot \log_{10}(N)$ complex multiplication and complex addition. Therefore, the larger the transform length is, the more obvious advantage the FFT method has.

![Diagram of implementation of 75-point FT](image)

Figure 1. The implementation of 75-point FT

The resource consumption of 80-point DFT, 128-point FFT, and 75-point FT is listed in Table 1. It can be seen that the 75-point FT method proposed in this paper has much less processing latency compared with 128-point FFT method, and consumes less multiplication resources compared with 80-point DFT method, which means it is an efficient implementation method of Fourier transform.

| DSP block          | LUT     | FF  | Latency($T_{clk}$) |
|--------------------|---------|-----|-------------------|
| 80-point DFT[9]    | 2560    | 233120 | 246080 | 6 |
| 128-point FFT IP [10] | 28 | -- | -- | 343 |
| 75-point FT        | 920     | 32610 | 17160 | 12 |

### 4. Simulation, implementation, and analysis

#### 4.1 Performance simulation

Assuming the input digital signal of Fourier transform is a single frequency signal, its carrier frequency $f_c = kf_s$, where $k$ is an integer between 0 and $N-1$, and the number of sampling points of this signal is 80, i.e., $M = 80$. In this simulation, 75-point DFT and mixed-radix 75-point FT, 80-point DFT, 128-point (zero padding) FFT, and 1200-point FFT are performed, respectively. The number 1200 is the least common multiple of 75 and 80, which is 16 times of 75 and 15 times of 80.

Set $k = 10$, the simulation results of 75-point DFT and mixed-radix 75-point FT are shown in Figure 2. It can be seen from the figure that the spectrum result of mixed-radix FT, which is based on Cooley-Tukey, Good-Thomas, and WFTA algorithm, is consistent with the result of 75-point DFT, which determines the correctness of mixed-radix FT method.

The simulation results of 1200-point DFT, 80-point DFT and 75-point FT are shown in Figure 3. It can be seen from Figure 3(a) that the spectrum results of 80-point DFT and 75-point FT are regularly consistent with the result of 1200-point DFT. The consistent spectrum points are extracted and plotted in Figure 3(b) and Figure 3(c), respectively. As we can see, the 75-point FT is equivalent to the 80-point DFT in this implementation, and each of them is an effective sample of the theoretical continuous spectrum of the input signal.
4.2 Spectrum Density

In this simulation, a Taylor window is utilized for a low side lobe spectrum and \( k \) takes the value of 0, 1, 2, ..., 10 in turn. The spectrum outputs corresponding to 75-point FT, 80-point DFT and 128-point FFT are depicted in Figure 4. As shown in the figure, the adjacent spectrums of 128-point FFT, 80-point DFT, and 75-point FT are overlapped at -1.009dB, -2.651dB, and -3.034 dB, respectively. The truth is that to cover the range from 0 to \( f_s \), the number of spectrum lines is 54, 57, and 91 for 75-point FT, 80-point DFT, and 128-point FFT, respectively, which means almost half of the spectrum lines is eliminated to use 75-point FT rather than 128-point FFT.
5. Conclusions

In this paper, we study the mixed-radix Fourier transform algorithm and extend the transform formula to the $N < M$ situation. An implementation example with $N=75$ and $M=80$ is described and analysed in detail. The results show that the mixed radix FT method has less resource consumption and less processing latency compared with FFT and DFT method, respectively and the proposed $N < M$ FT method increases the flexibility of traditional Fourier transform algorithm.

References

[1] REN Yongfeng, WU Haonan, Chu ChengQu, et al. 2D-FFT Image Edge Enhancement Design Based on FPGA [J]. JOURNAL OF GRAPHICS, 2019, 40(1):139-144.

[2] Yu C, Yen M H . Area-Efficient 128- to 2048/1536-Point Pipeline FFT Processor for LTE and Mobile WiMAX Systems[J]. IEEE Transactions on Very Large Scale Integration Systems, 2015, 23(9):1793-1800.

[3] Chen XuDong, Chen ZhangJin, Li HanChao, et al. Research and implementation of the spectrum analysis system based on FPGA [J]. ELECTRONIC MEASUREMENT TECHNOLOGY, 2016, 39(11):113-117.

[4] Zhang SheXin, Li WenChen, Yuan Xiang-yu, et al. The Research on the Formation of the Digital Beam by the Fourier Counterchange Method [J]. JOURNAL OF PROJECTILES, ROCKETS, Missiles and GUIDANCE, 2006, 26(4):255-258.

[5] Tukey C J W. An algorithm for the machine calculation of complex Fourier series [J]. Mathematics of Computation, 1965, 19(90):297-301.

[6] I.J.Good. The Interaction Algorithm and Practical Fourier Analysis [J]. Journal of the Royal Statistical Society. Series B: Methodological.1960, 20(2): 361-372.

[7] S. Winograd. On computing discrete Fourier transform, Proceedings of the National Academy of Sciences of the United States of America, 1976, 73(4):1005-1006;

[8] Garrido M, Gustafsson O, Qureshi F. Unified architecture for 2, 3, 4, 5, and 7-point DFTs based on Winograd Fourier transform algorithm[J]. Electronics Letters, 2013, 49(5):348-U60.

[9] Wu Chao. Design and implementation of FPGA DFT algorithm Based on IP [J]. Computer Knowledge and Technology, 2014, 10(20):4903-4905.

[10] Dou XiuMei, Zhao ZhenGang. Design and Implementation of FPGA FFT Algorithm Module Based on IP Core [J]. RADIO ENGINEERING OF CHINA, 2008, 38(8):29-31.