Research on High Precision FFT Algorithm Based on FPGA

Dandan Zhang
Shanghai Institute of Technology
Shanghai, China
zhdd_ang@163.com

Lan Chen
Shanghai Institute of Technology
Shanghai, China
chenlan@sit.edu.cn

Yajun Wu
Shanghai Astronomical Observatory
Shanghai, China
wuyajun@shao.ac.cn

2020 5th International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS) | 978-1-7281-8022-9/20/$31.00 ©2020 IEEE | DOI: 10.1109/ICIIBMS50712.2020.9336417

***Abstract*—In this paper, a high-precision Fast Fourier Transform algorithm is provided. Using the IP core provided in ISE, a software developed by Xilinx company, 16384 points of discrete data are calculated by Fast Fourier Transform. Using ISE, the simulation operation of FAST Fourier Transform of IEEE-754 floating point data and traditional fixed-point data was carried out, and the simulation results were compared and analyzed with those of MATLAB. In order to improve the precision of Fast Fourier Fransform, a comparative study is carried out from the aspects of operation time, resource consumption and precision. The simulation results show that the accuracy of IEEE-754 single precision floating-point simulation is significantly higher than that of fixed-point simulation. Therefore, IEEE 754 can greatly improve the operation accuracy of Fast Fourier Transform.**

Keywords—Fast Fourier Transform; IP Core; IEEE 754; FPGA; fixed-point arithmetic

I. INTRODUCTION

With the development of communication technology, digital signal processing technology plays a more and more important role in the field of astronomical observation and digital communication, and the requirements for signal processing accuracy are higher and higher. Among them, Fast Foutier Transform (FFT) is the core algorithm of digital signal processing, so it is of great significance to improve the accuracy of FFT algorithm[1].

This paper is based on the Field Programmable Gate Array (FPGA) to study the accuracy of FFT algorithm. FPGA is a customizable integrated circuit, which has a physical structure for digital signal processing algorithm[2]. FFT algorithm based on FPGA has the characteristics of flexibility, high efficiency and dynamic configuration[1]. In software programming, the ISE developed by Xilinx company is used. There are abundant IP cores inside, which can be customized according to the needs. It has great flexibility and saves the programming time.

978-1-7281-8022-9/20/$31.00 ©2020 IEEE

II. INTRODUCTION TO FFT ALGORITHM

FFT IP Core is provided in ISE software. In this study, FFT IP Core can be configured easily and quickly for programming.

A. *FFT IP Core Mode Selection*

The FFT core provides four architecture options, namely Pipelined, Radix-4, Radix-2 and Radix-2 Lite. For the data flow input form, the pipeline structure mode is adopted in this paper. The basic unit of pipeline structure is composed of Radix-2 butterfly algorithm, as shown in Figure 1.

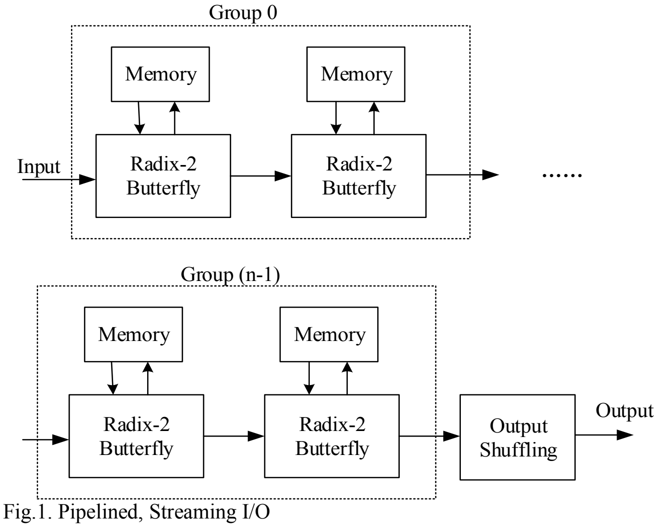


Fig.1. Pipelined, Streaming I/O

B. *Radix-2 Butterfly Algorithm*

FFT is a fast algorithm for Discrete Fourier Transform (DFT), The basic formula for the DFT algorithm are shown below:

$$X(k)=\sum_{k=0}^{N-1}x(n)e^{-j2\pi kn/N},\quad \mathrm{k}=0,1,2\ldots\ldots \qquad (1)$$

Divide the polynomial into odd and even parts:

$$\begin{cases} X(k)=X_1(k)+W_N^kX_2(k) \\ X(k+\frac{N}{2})=X_1(k)-W_N^kX_2(k) \end{cases} \qquad k=0,1,2\ldots\frac{N}{2}-1 \quad (2)$$

Where $W_N^k=e^{-j2\pi kn/N}$, the butterfly operation signal flow diagram is shown in Figure 2 below:

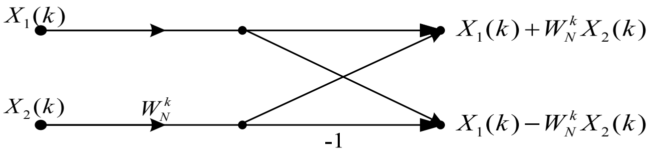


Fig.2. Flow chart of Radix-2 butterfly arithmetic

If $A^{'}=X(k)$, $B^{'}=X(k+\frac{N}{2})$, $A=X_1(k)$, $B=W_N^k$, $W=X_2(k)$, the formula (2) can be simplified into the following form (3):

$$\begin{cases} A^{'}=A+BW \\ B^{'}=A-BW \end{cases} \qquad (3)$$

Through comparison, it can be seen that the Radix-2 butterfly algorithm is to convert the original N point DFT into N/2 point DFT. It can greatly improve the operational efficiency[3].

III. Floating-Point analysis

Although the fixed-point algorithm has the advantages of fast speed and small occupancy area. In many cases, however, it is more advantageous to use floating point numerical format for mathematical calculations. In addition, designers must introduce many intermediate data types (with different fixed-point formats) to achieve the desired quality results when using fixed-point formats. By contrast, floating point formats can express real Numbers over a much wider range, making it easier for designers to use a single data type during programming[4]. Therefore, IEEE-754 single precision floating point is adopted to improve the accuracy of FFT calculation.

The data structure of the traditional fixed-point number is shown in Figure 3. It can be seen that the decimal point of the fixed-point number is fixed. If the length of the data exceeds the set precision during the operation, the operation result needs to be truncated, which will inevitably reduce the accuracy of the operation data. If the results are truncated at each level, the error of the results will be greater.

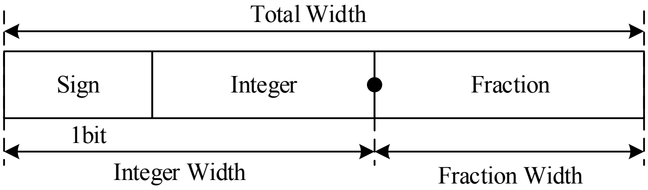


Fig.3. The structure diagram of fixed-point data

The structure of IEEE-754 single precision floating-point data is shown in figure 4, and its absolute value indicates that the precision is 1.8×10^{-38}~3.4×10^{-38} [5].

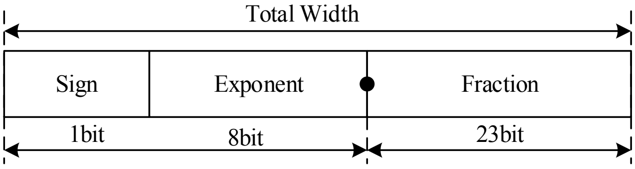


Fig.4. The structure diagram of IEEE-754 single precision floating-point data

In IEEE-754 single precision floating-point data, the position of the decimal point is not fixed, and the decimal point position is determined by the exponent bit, which can make full use of the last 23 decimal places and greatly improve the operation accuracy.

IV. Block diagram of FFT system

In this study, discrete fixed-point data is provided from outside, and fixed-point data is converted to single precision floating-point data through the Floating-Point IP Core in ISE, and uses Floating-Point Numbers for FFT. The FFT system block diagram is shown in Figure 5.

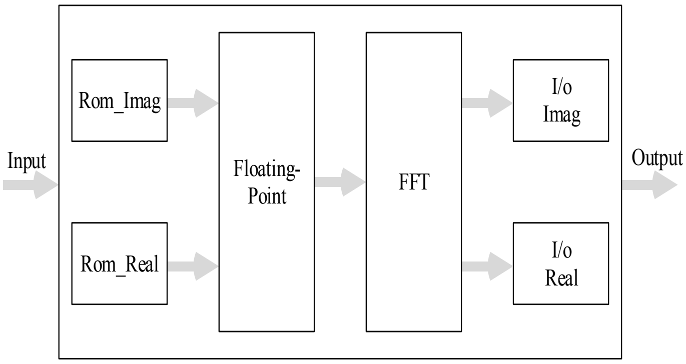


Fig.5. Block diagram of FFT system

V. System simulation

The sine signal is generated by MATLAB, and 16384 discrete signals are generated by sampling and adding Gaussian white noise. Suppose that the baseband signal of the sine signal is f0, the sampling frequency is fs, and the Gaussian white noise is awgn. In this paper, baseband signal f_0=1024(HZ), sampling frequency fs=16384(HZ) and Gaussian white noise awgn=200(dB) illustrated as examples. Figure 6 is the sine signal diagram with Gaussian white noise.

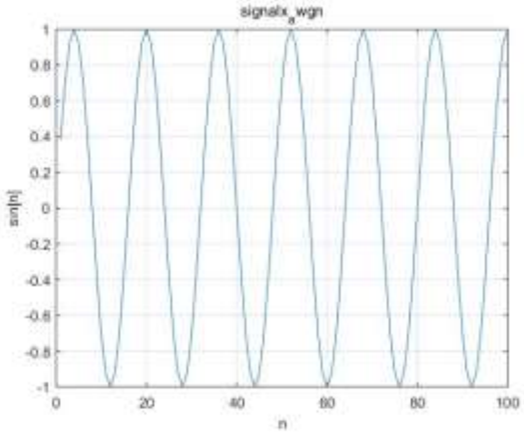
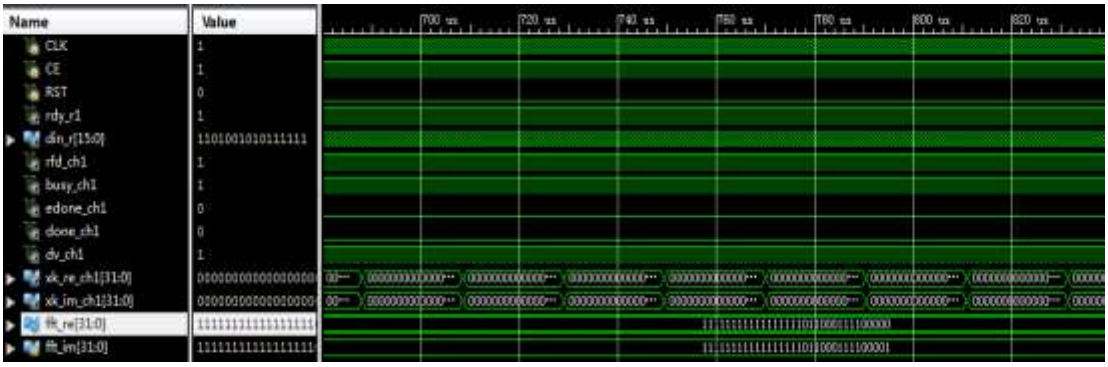


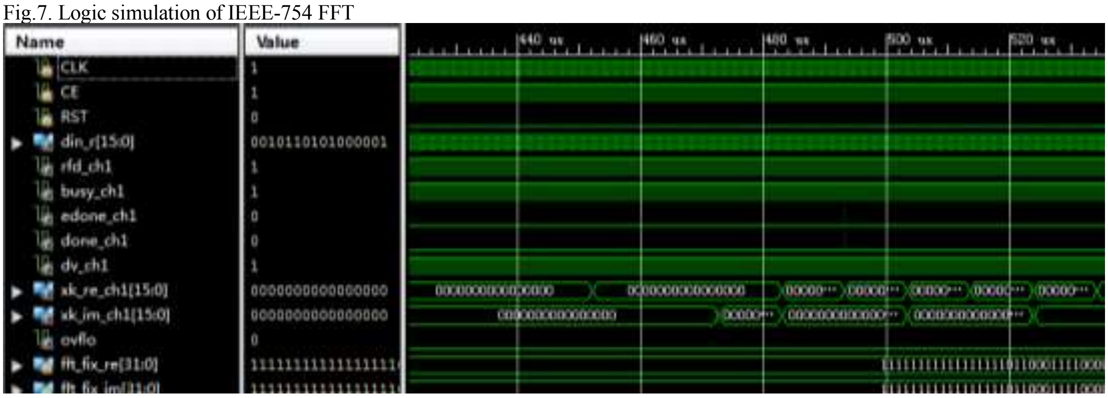
Fig.6. Graph of sine and Gaussian white noise signal

The discrete signal data generated by software MATLAB is rounded to truncate, and 16384 16 bit discrete points are obtained. 16384 discrete points were written into *.coe file and imported into ROM IP core. Then, 16384 IEEE-754 single-precision Floating point datas are obtained by processing the fixed-point data with Floating-point IP core. Finally, FFT IP core is used to verify the 16384 point discrete signal fixed-point datas and IEEE-754 single precision floating-point datas for Fast Fourier Transform. Then, the analysis, synthesis and logic simulation are respectively carried out. The two logic simulation waveforms are shown in Figure 7 and Figure 8.



```
Name             Value
CLK              1
CE               1
RST              0
rdy_r1           1
din_r[15:0]      1101001010111111
rfd_ch1          1
busy_ch1         1
edone_ch1        0
done_ch1         0
dv_ch1           1
xk_re_ch1[31:0]  0000000000000000
xk_im_ch1[31:0]  0000000000000000
fft_re[31:0]     1111111111111111
fft_im[31:0]     1111111111111111
```

Fig.7. Logic simulation of IEEE-754 FFT



```
Name             Value
CLK              1
CE               1
RST              0
din_r[15:0]      0010110101000001
rfd_ch1          1
busy_ch1         1
edone_ch1        0
done_ch1         0
dv_ch1           1
xk_re_ch1[15:0]  0000000000000000
xk_im_ch1[15:0]  0000000000000000
ovflo            0
fft_fix_re[31:0] 1111111111111111
fft_fix_im[31:0] 1111111111111111
```

Fig.8. Logic simulation of fixed-point FFT

In Figure 7 and 8, din_r is 16 bit discrete point in the ROM IP core imported, and xk_re_ch1 and xk_im_ch1 are the spectrum data generated by channel 1 after the discrete data is calculated by FFT IP core, which are divided into real part and imaginary part. When FFT IP core performs Fast Fourier Transform on discrete data, the flag bit busy_ch1 is high level. When dv_ch1, the effective marker of output data, is high, the simulation spectrum data will be output and recorded respectively.

The spectrum diagram of ISE simulation results and MATLAB simulation results are drawn respectively, as shown in Figure 10, Figure 11 and Figure 12 below. The spectrum data of ISE simulation output is compared with the spectrum data of MATLAB software.

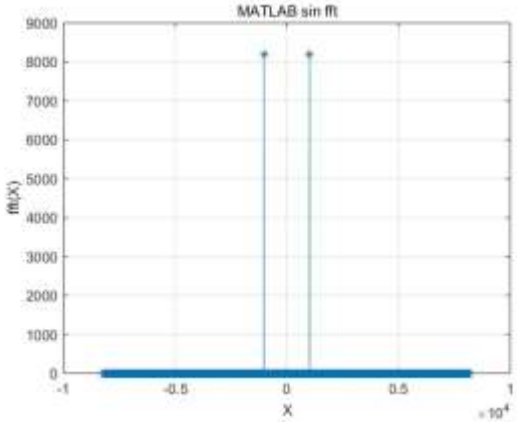


Fig. 9. MATLAB FFT spectrum diagram

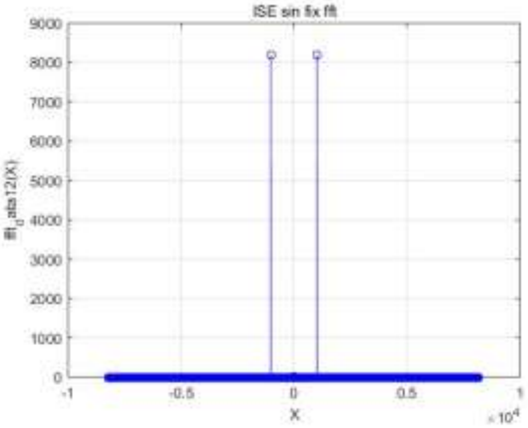


Fig. 10. ISE IEEE-754 FFT spectrum diagram

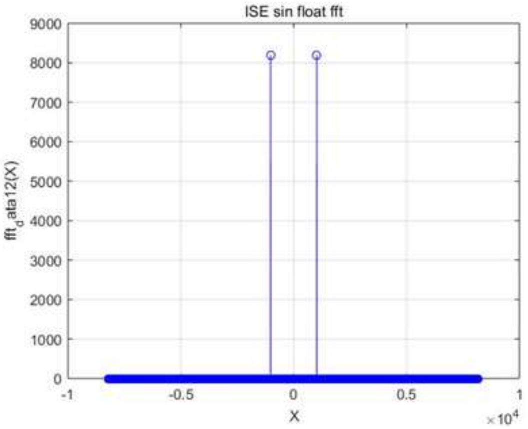


Fig. 11. ISE Fixed-Point FFT spectrum diagram

The simulation result of FFT with MATLAB is 8192.005416149652, the simulation result of fixed-point FFT with ISE is 8192.539288889617, and the simulation result of IEEE-754 single precision floating-point FFT with ISE is 8192.005569347877. By comparison, the spectrum accuracy of FFT using IEEE-754 single precision floating-point is obviously higher. Table 1 below shows the comparison of simulation results after FFT of 16384 discrete signals under the conditions that the sampling frequency f_s=16384(HZ), Gaussian white noise and different baseband signals.

The mean deviation analysis of simulation results is carried out. Assume that the frequency spectrum datas of the Fast Fourier Transform obtained by MATLAB are $(a_0, a_1, \ldots, a_n)$. The frequency spectrum datas obtained by taking the Fast Fourier Transform of IEEE-754 floating-point numbers using ISE are $(b_0, b_1, \ldots, b_n)$. Mean deviation calculations are made for the 10 groups of MTLAB spectrum data in Table 1 and the spectrum data calculated by IEEE 754 single-precision floating point method, as shown in Formula (4).

$$\bar{\theta}_0 = \frac{(|a_0 - b_0| + |a_1 - b_1| + \ldots + |a_9 - b_9|)}{10} \tag{4}$$

The mean deviation $\bar{\theta}_0$= 0.000206824163888086.

And similarly, assume the frequency spectrum datas by taking the Fast Fourier Transform of fixed-point numbers using ISE are $(c_0, c_1, \ldots, c_n)$. Calculate the mean deviation of 10 groups of MTLAB spectrum data in Table 1 and the spectrum number calculated by ISE fixed-point method, as shown in Formula (5).

$$\bar{\theta}_1 = \frac{(|a_0 - c_0| + |a_1 - c_1| + \ldots + |a_9 - c_9|)}{10} \tag{5}$$

The mean deviation $\bar{\theta}_1$= 0.859711820967823.

According to formula $|\bar{\theta}_0 - \bar{\theta}_1| / \bar{\theta}_1$, it can be calculated that the IEEE-754 single-precision floating point algorithm improves the frequency spectrum accuracy of Fast Fourier Transform by 0.999759426171836.

TABLE I. THE COMPARISON OF SPECTRUM ACCURACY

f_0(HZ)	MATLAB	ISE IEEE 754	ISE Fixed Point
256	8191.973892071450	8191.973684290826	8192.955022456794
371	8192.005733846199	8192.005837034247	8192.881056136479
512	8191.959974237958	8191.959852637964	8193.098070937513
908	8192.020940449904	8192.021186782607	8192.576639861234
1024	8192.005416149652	8192.005569347877	8192.539288889617
1766	8192.011712866237	8192.012212614538	8191.312471148931
2048	8191.916030023076	8191.916305190005	8190.000244200241
3152	8192.028769840503	8192.028839358196	8192.467271829653
4096	8192.000000000000	8192.000000000000	8192.000000000000
6521	8192.005733846199	8192.006125555325	8193.465201000125

VI. PERFORMANCE COMPARISON

According to the data results of ISE simulation, using IEEE-754 to perform FFT can improve the accuracy of data operation, and the resource consumption is higher than fixed-point operation. The values of FFT performance parameters of fixed-point data and floating-point data are determined in the process of IP Core configuration. The comparison of parameters is shown in Table 2.

In Table 2, the relative minimum principle is used for the phase factors of fixed-point and floating-point operations. The comparison of performances is shown in Table 3.

From the data in the Table 3, it can be seen that the single precision floating-point FFT using IEEE-754 standard data is slightly higher than the fixed-point operation in terms of operation speed and resource occupation, but it can be easily realized in the existing technology. In the field where the accuracy of digital signal processing results is high, the IEEE-754 single precision floating-point operation is fully available.

TABLE II. THE COMPARISON OF PARAMETERS

Parameter	IEEE 754	Fixed Point
Implementation	Pipeliined,Streaming I/O	Pipeliined,Streaming I/O
Target Clock Frequence(MHZ)	256	256
Transform Size	16384	16384
Input Data Width	16	16
Output Ordering	Natural Order	Natural Order
Phase Factor Width	24	16

TABLE III. THE COMPARISON OF PERFORMANCES

Performance	IEEE 754	Fixed Point
Transform Cycles	65712	49296
XtremeDSP	54	18
18K Block RAMS	197	68
Latence（μs）	256.688	192.563

VII. CONCLUSION

FFT algorithm is an important operation in digital signal processing. It is widely used in radar, observation, tracking, high-speed image processing, secure wireless communication and digital communication[7]. This design is based on FPGA experiments, using FFT algorithm, respectively, fixed-point operation and IEEE-754 single precision floating-point operation for comparative study. The results show that IEEE-754 single precision floating-point operation can effectively improve the accuracy of FFT, and through comparing with the parameters about performance of traditional fixed-point operation, it is found that under the existing technical conditions, the algorithm can be fully applied in various communication fields, and has a wide range of application value.

REFERENCES

[1] Wei Wang, Shuguo Liang, Xiaobo Ma, “FPGA implementation of high speed 65536 point FFT,” China integrated circuit, vol. 245, pp. 60-65, 2019

[2] Daxi Li, “Research on implementation of configurable FFT IP core based on FPGA,” Electronic SCI. & tech, vol. 27(6), pp. 46-53, 2014

[3] Guodong Liu, Boxiao Chen, Duofang Chen, “FPGA design of FFT,” Aviation computing technology, vol. 34(3), pp. 101 - 104, 2004

[4] James Hrica, “The floating point design is realized by Vivado HLS,” Electronic Publishing China, vol. (1), pp. 33-38, 2015

[5] Yu Rong, En Zhu. “Design of a high performance FFT butterfly computing unit ,” Journal of Southeast University, vol. 37(4), pp. 565-568, 2007

[6] Guangshu Hu, “Digital signal processing,” Beijing: Tsinghua University Press, 2005

[7] Xing Yang, Zhiyuan Xie, Li Rong, “Design of FFT processor based on FPGA,” International Electronic Elements, vol. (5), pp. 25-28, 2008