International Journal of Computational Intelligence Systems, Vol. 4, No. 6 (December, 2011), 1131-1139

Implementing EW Receivers Based on Large Point Reconfigured FFT on FPGA Platforms

He Chen, Xiujie Qu*, Yuedong Luo
School of Information and Electronic Engineering, Beijing Institute of Technology
Beijing, 100081, PR China

Chenwei Deng
School of Computer Engineering, Nanyang Technological University, 639798, Singapore

Received 20 August 2011; accepted 25 November 2011

Abstract

This paper presents design and implementation of digital receiver based on large point fast Fourier transform (FFT) suitable for electronic warfare (EW) applications. When implementing the FFT algorithm on field-programmable gate array (FPGA) platforms, the primary goal is to maximize throughput and minimize area. This algorithm adopts two-dimension, parallel and pipeline stream mode and implements the reconfiguration of FFT's points. Moreover, a double-sequence-separation FFT algorithm has been implemented in order to achieve faster real time processing in broadband digital receivers. The performance of the hardware implementation on the FPGA platforms of broadband digital receivers has been analyzed in depth. It reaches the requirement of high-speed digital signal processing, and reveals the designing this kind of digital signal processing systems on FPGA platforms.

Keywords: digital receivers, field programmable gate array (FPGA), fast Fourier transform (FFT), large point reconfigured, signal processing system.

1. Introduction

As current ultra deep submicron technologies developing, the design process of digital systems becomes more complicated, especially for application specific integrated circuit (ASIC). Contrasting with ASICs, FPGAs offer significant advantages at a suitable low cost. First, the designers can modify the implementations at any time. Second, the verification of a design mapped into an FPGA is very simple. Finally, the performance they can provide is better than general purpose CPUs or DSP, even though they are not as efficient as ASICs in terms of performance, area or power. So it makes FPGAs an attractive choice for offload computationally intensive digital signal processing functions from the processors[1].

In electronic warfare digital receivers, wideband frequency coverage, high sensitivity and dynamic range, high probability of intercept, simultaneous signal detection, excellent frequency resolution, and real time operation, are our requirements[2]-[6]. The wideband digital receivers mainly based on FFT, require intensive computation for real time applications. We can realize high-speed and real-time digital signal process on one chip FPGA, then the system throughput of many signal processing algorithms can be improved by exploiting concurrency in the form of parallelism and pipelining[7]-[11].

In this paper, we present an in-depth study on the hardware implementation of a large point reconfigured

[*] Corresponding author: quxiujie@bit.edu.cn

FFT. The pipelined and parallel architectures of this FFT are suitable for wideband EW digital receivers. The maximum sequence length of FFT can be 1M(2^{20}) points. Our purpose is to provide not only the hardware FFT architectures on one FPGA chip, but also the analysis of important parameters involved in hardware implementation on FPGAs, and to study the performance of the system.

The design concept of this paper can be applied to other fields, such as intelligent transport systems[12][13], radar signal processing systems.

2. The FFT Algorithm

2.1. *Holistic design principle*

For wideband signal process, we consider the signals process speed firstly. The array structure's size is too larger to realize and the recursion structure's process speed is too slow to reach real processing demand of the system. Although the parallel structure can obtain the high speed, it uses more memory units and operation units so that it can't be realized on one FPGA chip.

Bergland presents a two-dimension parallel arithmetic of FFT and this arithmetic can transform a long FFT to several short FFT so that it can reduce the operation amounts. For *N=L×M*, the *N* points DFT that we need to do is shown followed:

$$X(k)=\sum_{n=0}^{N-1}x(n)W_N^{kn} \quad . \tag{1}$$

where, *X(k)* is a frequency spectrum of input *x(n)*, $W_N^{nk}=\exp(-j2\pi nk/N)$ ($k=0,1,\cdots,N-1$) is called twiddle factor of DFT.

On the assumption that *N* is a complex number, and *N= L×M*, then *n* and *k* can be expressed by followed formula separately:

$n = Mn_1 + n_0$

where $n_1=0,1,\dots,L-1$; $n_0=0,1,\dots,M-1$;

$k = Lk_1 + k_0$

where $k_1=0,1,\dots,M-1$; $k_0=0,1,\dots,L-1$;

We can obtain the following formula:

$$\begin{aligned} X(k) &= X(Lk_1+k_0) = X(k_1,k_0) \\ &= \sum_{n=0}^{N-1} x(n)W_N^{kn} \\ &= \sum_{n_0=0}^{M-1}\left\{\left[\sum_{n_1=0}^{L-1} x(n_1,n_0)W_L^{n_1k_0}\right]W_N^{n_0k_0}\right\}W_M^{n_0k_1} \end{aligned} \quad . \tag{2}$$

By this formula, we can obtain the method of transforming the one-dimension large points FFT to two-dimension small points sub FFT:

Support *N=1024×4^n^, n=0,1,2,3,4,5*

Here we use radix-4 FFT arithmetic to transform one-dimension process to two-dimension process.

Step1: To rearrange data. Here *N*-points data are rearranged to *1024×4^n* format.

Step2: To do radix-4 FFT for every column data. It means that we should do *4^n* times 1024 point's one-dimension FFT.

Step3: To multiply the output of the column transformation by twiddle-factors, then save the middle data to the array which row is 4n and column is 1024.

Step4: To do radix-4 FFT for middle array's every row. It means that we should do 1024 times *4^n* point's one-dimension FFT.

Step5: To order the sequence by norm order or reverse order.

Because we choose the points to *1024×4^n*, here *n=0,1,2,3,4,5*, the column transformation is *L=1024* point's FFT. The point of column transformation is fixed. The row transformation is *M=4^n* point's FFT. The point of row transformation is variable.

Based on the analysis above, the system is made up of two parts shown in following Fig.1:

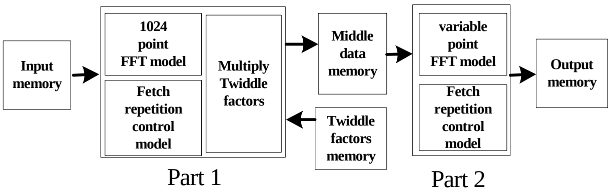

Fig.1 Large points and reconfigured FFT system

The two parts' main function modules are both FFT arithmetic. In each part, except FFT module, there are fetch data module, repetition control model. The FFT module completes the column transformation and the row transformation's butterfly operation. The fetch data module and the repetition control module complete these functions such as producing address, fetching data, reversing order, reconfiguring points and output data.

2.2. *The design of fixed points FFT*

The design of fixed points FFT adopts five level pipeline structure which uses decimation in time (DIT) radix-4 algorithm. Each level adopts ping-pong memories to realize pipeline. In order to avoid data overflow and obtain higher precision, we use block float algorithm. Radix-4 FFT algorithm includes pipeline unit, producing address unit, producing twiddle-factors address unit, butterfly operation unit and block float process unit. This is shown in Fig.2.

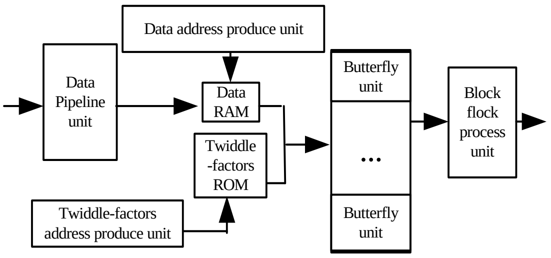

Fig.2 Fixed points FFT system

The equation of butterfly operation in DIT radix-4 FFT is as follows:

$$\begin{cases} X(n)=x(n)+x(n+4^{p-s})W_{4^s}^{n}+x(n+2\cdot 4^{p-s})W_{4^s}^{2n}+x(n+3\cdot 4^{p-s})W_{4^s}^{3n} \\ X(n+4^{p-s})=x(n)-jx(n+4^{p-s})W_{4^s}^{n}-x(n+2\cdot 4^{p-s})W_{4^s}^{2n}+jx(n+3\cdot 4^{p-s})W_{4^s}^{3n} \\ X(n+2\cdot 4^{p-s})=x(n)-x(n+4^{p-s})W_{4^s}^{n}+x(n+2\cdot 4^{p-s})W_{4^s}^{2n}-x(n+3\cdot 4^{p-s})W_{4^s}^{3n} \\ X(n+3\cdot 4^{p-s})=x(n)+jx(n+4^{p-s})W_{4^s}^{n}-x(n+2\cdot 4^{p-s})W_{4^s}^{2n}-jx(n+3\cdot 4^{p-s})W_{4^s}^{3n} \end{cases} .(3)$$

The butterfly operation unit is a duplicate unit in the design. Every level pipeline can call this unit to make the program modularized. It also makes the program easy to read. At the same time, we can easily to modify program by using it.

2.3. *The design of variable points FFT*

The variable points FFT design adopts the similar method as fixed point FFT. The difference between these two modules is that there is a configuration logic unit to control the operation in every level in variable points FFT.

If we do 1024 points FFT, we start up all five level operation units. If we do 256 points FFT, we start up ahead four level operation units and do not use the last level operation unit. Else points FFT adopts the same method to realize.

The variable points FFT is mainly composed by five units. They are data input unit, 64 points pipeline process unit, multiplying twiddle-factors unit, 4 points and 16 points pipeline process unit and data output unit. Every unit all adopts ping-pong memories to guarantee the real time operation of input data. The two sub FFT modules adopt parallel pipeline structure to improve the process speed.

The variable points FFT system is shown as Fig.3.

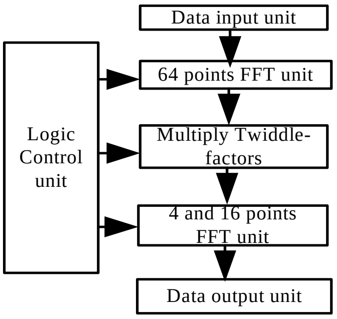

Fig.3 Variable points FFT system

2.4. *Twiddle factor memory compression*

From Eq. (2) and Fig.1 we can know that output sequence of the first 1024-point FFT ought to multiply twiddle factors $W_N^{n_0k_0}$. For computation of 1M-point FFT the storage capacity of twiddle factor $W_N^{n_0k_0}$ is 1023×1023 words, so that one single FPGA or ASIC can't load it. The common method is to store a part of twiddle factors using symmetric feature of sine and cosine which can compute others twiddle factors.

In this paper based on numerical value feature of twiddle factor one linear interpolation method is presented to use less twiddle factors to compute all 1M point twiddle factors.

First let us analyze the numerical value curve of twiddle factor. We can see from Eq. (1) that the twiddle factor W_N^i can be regarded as the combination of the evenly sampled sine and cosine. Supposing $R'(i)=\cos(2\pi\cdot i/N)$ and $I'(i)=\sin(2\pi\cdot i/N)$, then sine value $I'(i)$ have linear relation with *i*, as well as cosine value $R'(i)$, so the trigonometric functions curve can be divided into segments.

Sequential P $R'(i)$s and $I'(i)$s make a group, each group may be approximated by a line segment, so:

$$\begin{gathered} R'(i) \approx R_c(i) = R_{Bk} + R_{\Delta k} \cdot (i - i_k) \\ I'(i) \approx I_c(i) = I_{Bk} + I_{\Delta k} \cdot (i - i_k) \\ i_k \le i < i_{k+1} \end{gathered} . \quad (4)$$

Where R_{Bk} and I_{Bk} are start-point of number k line segment, $R_{\Delta k}$ and $I_{\Delta k}$ are the slopes of the line segments.

So, we may only store R_{Bk} and I_{Bk}, $R_{\Delta k}$ and $I_{\Delta k}$ instead of all W_N^i, other factors can be calculated by one multiplication and one addition.

To simplify control P may usually be the power of 2, namely $P = 2^m$.

For ultra-long FFT this method fits to compress of $W_{1024\times1024}^{n_0 k_0}$ in Fig.1 because high compression ratio can be obtain by bigger P. For instance, the storage twiddle factor of 1024×1024 points FFT can be compressed to 2k×32 bits from 2M×18 bits when P=128 and bit-length of twiddle factor is 18, precision can be ensured simultaneity.

2.5. *Double sequence separation*

Considering input data are real sequence in practical application, this part introduces how to implement two FFTs of N-point real sequence with one complex FFT. On the assumption that $x_1(n)$ and $x_2(n)$ are two real sequences of length N, then complex sequence $x(n)$ can define as follows:

$$x(n) = x_1(n) + jx_2(n) \qquad 0 \le n \le N-1 . \quad (5)$$

Because of linear characteristic of DFT, DFT of $x(n)$ can express as:

$$X(k) = X_1(k) + jX_2(k) . \quad (6)$$

Considering $x_1(n)$ and $x_2(n)$ can be written function of $x(n)$:

$$x_1(n) = \frac{x(n) + x^*(n)}{2} . \quad (7)$$

$$x_2(n) = \frac{x(n) - x^*(n)}{2j} . \quad (8)$$

So DFT of $x_1(n)$ and $x_2(n)$ are：

$$X_1(k) = \frac{1}{2}\{DFT[x(n)] + DFT[x^*(n)]\} . \quad (9)$$

$$X_2(k) = \frac{1}{2j}\{DFT[x(n)] - DFT[x^*(n)]\} . \quad (10)$$

Because of

$$\begin{aligned} DFT[x^*(n)] &= X^*((-k))_N R_N(k) \\ &= X^*((N-k))_N R_N(k) = X^*(N-k) \cdot \\ &0 \le k \le N-1 \end{aligned} \quad (11)$$

So we can obtain:

$$\begin{aligned} X_1(k) &= \frac{1}{2}[X(k) + X^*(N-k)] \\ X_2(k) &= \frac{1}{2j}[X(k) - X^*(N-k)] \end{aligned} . \quad (12)$$

In order to deduct formula of FPGA implementation, above complex expressions are expanded to real representation.

If $X(k) = R_k + jI_k$ and $X(N-k) = R_{N-k} + jI_{N-k}$, above formula can be:

$$\begin{aligned} X_1(k) &= \frac{1}{2}[X(k) + X^*(N-k)] = \frac{1}{2}(R_k + jI_k + R_{N-k} - jI_{N-k}) \\ &= \frac{1}{2}(R_k + R_{N-k}) + j\frac{1}{2}(I_k - I_{N-k}) \end{aligned} .(13)$$

$$\begin{aligned} X_2(k) &= \frac{1}{2j}[X(k) - X^*(N-k)] = \frac{-j}{2}(R_k + jI_k - R_{N-k} + jI_{N-k}) \\ &= \frac{1}{2}(I_k + I_{N-k}) + j\frac{1}{2}[(R_{N-k} - R_k)] \end{aligned} .(14)$$

3. Performance Analysis of Receiver

It might be important to note that the performance on false alarm probability after FFT. By researching the algorithm of threshold generating of pulsed radar signals' spectrum detection in clutter background, we have presented a threshold generating algorithm based on the structure of CMLD (Censored Mean Level Detector) CFAR detector, and analyze false alarm rate.

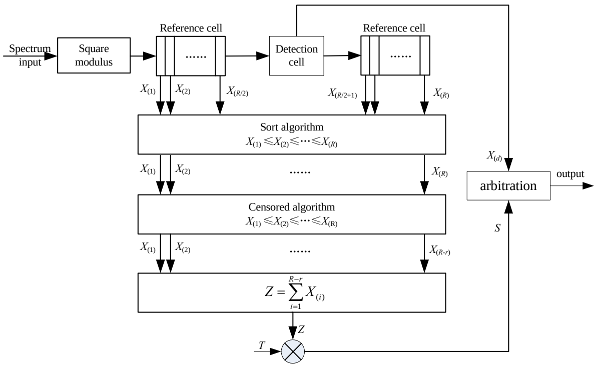

Fig.4 The CMLD-CFAR detection algorithm

After DFT, *N*-point time domain sequence *x(n)* is transformed to frequency domain *N*-point *X(k)*, assuming the location of spectrum peak (the carrier wave frequency of pulse radar signal) is k_0, the left harmonic's location is k_i. Due to Gauss distribution random variable still follows Gauss distribution after linear transformation, *X(k)* still follows Gauss distribution. The envelop of *X(k)* follows Rayleigh distribution:

$$p_n(x) = \frac{x}{\sigma^2}\exp\left(-\frac{x^2}{2\sigma^2}\right) \quad . \qquad (15)$$

σ is the mean square of Gauss distribution. The output of $X(k_i)$ follows Rice distribution every location of harmonic. Assuming that *I* is the power ratio of the signal and background clutter before DFT, and then after DFT, *NI* is the power ratio of the carrier wave whose frequency is k_0 and background clutter. The mod of $X(k_0)$ follows Rice distribution, its PDF (Probability Density Function) is

$$p_s(x) = \frac{x}{\sigma^2}\exp\left(-\frac{x^2}{2\sigma^2} - NI\right) I_0\left(\frac{x}{\sigma}\sqrt{2NI}\right) \quad . \qquad (16)$$

When the SCR (Signal-to-Clutter Ratio) is smaller, the above Rice distribution tends to follow the Rayleigh distribution, and when the SCR is bigger, it tends to follow Gauss distribution. Only the situation that the SCR is smaller will be discussed below.

After the square-law detection is done to *X(k)*, the output follows exponential distribution.

$$p_D(x) = \frac{1}{\lambda'} e^{-x/\lambda'} \quad x \geq 0 \quad . \qquad (17)$$

In Eq.(17)

$$\lambda' = \begin{cases} \mu & H_0 \\ \mu(1+\lambda) & H_1 \end{cases} \quad . \qquad (18)$$

μ is the total average power of the background clutter and thermal noise, λ is the ratio of the target-signal average power and the clutter-noise power. H_0 is the assumption of none target, H_1 is the assumption of having target. When there is no target, clutter noise is symmetrical, *X(k)* is statistical independent but follows the same distribution.

If *S=TZ* is the detection threshold value, it is a random variable itself, so we can use *S*'s statistical character to figure out the false alarm probability:

$$\begin{aligned} P_{fa} &= E_S\left\{P\left[\left|X(k)\right|^2 \geq S \middle| H_0\right]\right\} \\ &= \int_0^\infty f_Z(z)\int_{Tz}^\infty \frac{1}{\mu}e^{-x/\mu}dxdz = M_z(u)\,|_{u=T/\mu} \end{aligned} \quad . \qquad (19)$$

$f_Z(z)$ is the PDF of *Z*, $M_Z(u)$ is the MGF (Moment Generating Function) of the random variable *Z*. So the detection probability under symmetrical clutter background is

$$P_d = M_z(u)\,|_{u=T/\mu(1+\lambda)} \, . \qquad (20)$$

X(i),i=1,2,⋯R is the sequential sampling in Eq.(1) , they are no longer independent after linear transformation

$$Z = \sum_{i=1}^{R-r} X_{(i)} = \sum_{i=1}^{R-r} v_i \quad . \qquad (21)$$

where, $v_i = (R-r-i+1)(X_{(i)} - X_{(i-1)}),\ X_{(0)} \equiv 0$. So $V_i(i=1,2\cdots R\text{-}r)$ are statistically independent, its PDF is:

$$f(v_i) = \frac{c_i}{\mu} e^{-c_i v_i/\mu} \quad . \qquad (22)$$

In Eq. (22)

$$c_i = \frac{R\text{-}i+1}{R\text{-}r\text{-}i+1} \quad . \qquad (23)$$

Then, MGF of v_i is

$$M_{v_i}(u)=\frac{c_i}{\mu u+c_i} \quad . \qquad (24)$$

MGF of Z in Eq. (21) is the product of $M_{v_i}(u)\ (i=1,2\cdots R-r)$.

$$M_Z(u)=\prod_{i=1}^{R-r}\frac{c_i}{\mu u+c_i} . \qquad (25)$$

Thinking about the relationship between PDF and MGF, the PDF of Z is

$$\begin{aligned} f_Z(z) &= LT^{-1}\left\{\prod_{i=1}^{R-r}\frac{c_i}{mu+c_i}\right\} \\ &= LT^{-1}\left\{\sum_{i=1}^{R-r}\frac{c_i}{mu+c_i}\left[\prod_{\substack{j^1 i\\ j=1}}^{R-r}\frac{c_j}{c_j-c_i}\right]\right\} \end{aligned} . \qquad (26)$$

LT^{-1} represents the Laplace transformation. Define a_i as:

$$\begin{aligned} a_i &= \frac{\prod_{j=1}^{R-r} c_j}{\prod_{\substack{j\neq i\\ j=1}}^{R-r} c_j-c_i} \\ &= \binom{R}{r}\binom{R-r}{i-1}(-1)^{i-1}\left(\frac{R-i+1-r}{r}\right)^{R-r-1} \end{aligned} . \qquad (27)$$

So the PDF and MGF of Z are:

$$f_Z(z)=\sum_{i=1}^{R-r}\frac{a_i}{\mu}e^{-c_i z/\mu} . \qquad (28)$$

$$M_Z(u)=\sum_{i=1}^{R-r}\frac{a_i}{c_i+\mu u} \quad . \qquad (29)$$

Review the Eq.(19) and Eq.(20), the detection probability and false alarm rate of CMLD-CFAR in symmetrical clutter background are that:

$$P_d=\sum_{i=1}^{R-r}\frac{a_i}{c_i+T/(1+\lambda)} \quad . \qquad (30)$$

$$P_{fa}=\sum_{i=1}^{R-r}\frac{a_i}{c_i+T} \quad . \qquad (31)$$

Calculating the threshold of GO-CFAR and CMLD-CFAR separately, we can choose the bigger threshold:

$$l=\mathrm{MAX}(l_{\text{GO-CFAR}},l_{\text{CMLD-CFAR}}) . \qquad (32)$$

Fig.5 shows the simulation of the above-mentioned method. Simulation conditions include of two signals' that frequencies are 125.0MHz and 127.6MHz, Gauss random noise, $\mu=0$, $\sigma^2=5$; $N=256K$-point DFT; GO-CFAR threshold parameter: reference cell $M=64$, number of defense cell is 8; CMLD-CFAR threshold parameter: $R=64$, $r=8$, nominal factor $T=0.125$.

As is shown in Fig.5, on condition that it has larger clutter background, the CMLD-CFAR should be chosen as threshold. During the frequency domain detection, the harmonic frequency points are correlative with the signal, they should not be reckoned into the noise power, and moreover, CMLD-CFAR just fulfills this requirement. When the threshold is built by CMLD-CFAR and the nominal factor T is chosen, it is deduced from Eq.(31) that detection false alarm rate P_f is invariable. When it is to GO-CFAR, ensuring the same detection probability, the detection false alarm rate is less than P_f; this is helpful to the detection.

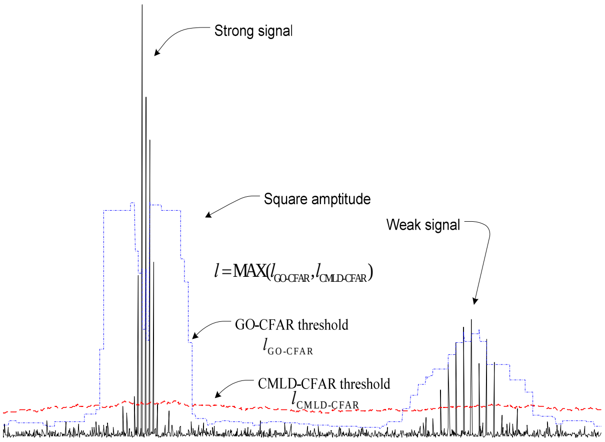

Fig.5 Simulation of the threshold building

The algorithm can obtain stable estimation of spectrum clutter background and provide the conditions the algorithm suits. The academic analysis and the simulation prove the algorithm’s validity.

4. FFT-Based Digital Receiver

There are many additional elements being required, even though the basic implementation of wideband digital receivers is based on FFT algorithm. This section is devoted to the analysis and design of the whole system, because the configuration and implementation of all the elements involved can significantly influence the final performance of the receiver. In this sense, CFAR threshold, windowing or frequency detection may play an important role in the system because some FPGA resources may be required for its implementation.

Fig.6 illustrates the FFT spectrum detection modules of wideband digital receivers. They have implemented on one FPGA chip.

For a test signal which make up of four channels’ different pulse radar signal as is shown in Tab.1. The signal has been processed by the digital receiver’s digital processing part, the FFT results are shown in Fig.7.

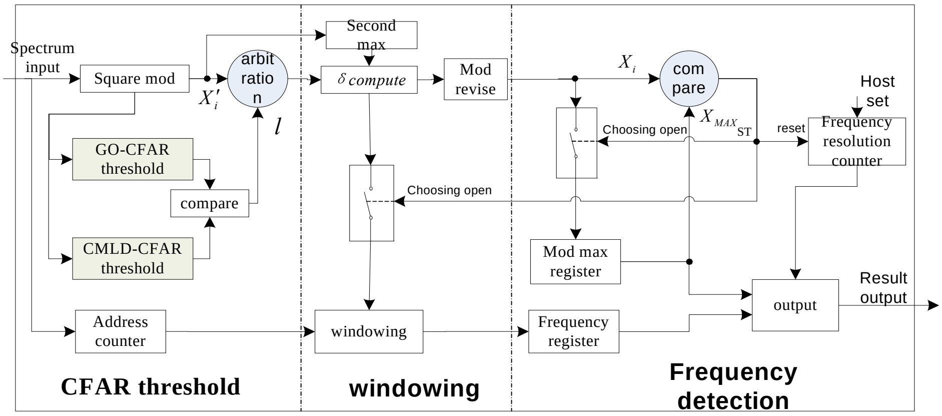

Fig.6 FPGA implementation of wideband digital receiver

Table 1. Parameters of four channels’ different signals

Signal channel	RF/MHz	TOA/us	PRI/us	PW/us
1	52.000	0.000	10.000	1.000
	200.000	0.000	7.000	2.000
	79.000	0.000	8.000	3.000
	166.000	1.000	9.000	1.000
2	60.000	0.000	10.000	1.000
	212.000	1.000	10.000	1.000
	89.000	2.000	8.000	2.000
	153.000	3.000	7.000	2.000
3	73.000	1.000	10.000	1.000
	136.000	0.000	9.000	1.000
	185.000	2.000	8.000	1.000
	238.000	3.000	10.000	1.000
4	37.000	1.000	10.000	1.000
	167.000	2.000	10.000	2.000
	196.000	3.000	10.000	3.000
	221.000	0.000	10.000	1.000

The system can detect four max signals from the 4 channel signal, and the detecting results of system are shown in Tab.2. We also can find the four max signal from the FFT results, they are the RF=98MHz signal from channel 1, RF=153MHz signal from channel 2, RF=196MHz signal from channel 4, RF=89MHz signal from channel 2.

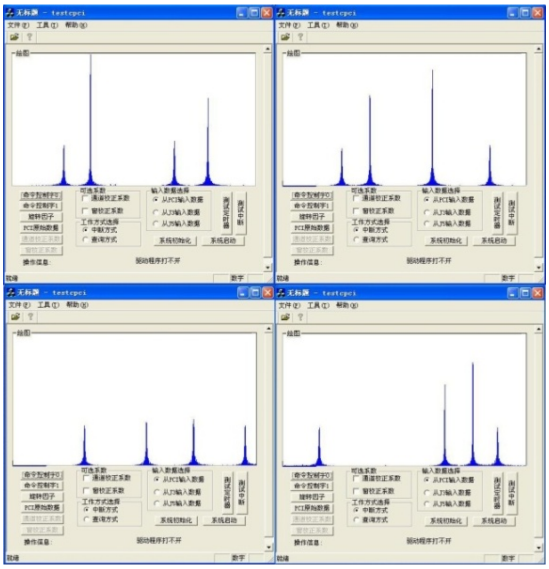

Fig.7 FFT results for the test signal

Table 2. Detecting Parameters of four channels' different signal by system

Signal channel	RF/MHz	TOA/us	PRI/us	PW/us
1	78.980	0.000	7.992	2.976
2	152.968	3.000	7.000	2.000
4	195.860	3.000	10.000	3.000
2	88.855	1.992	8.000	2.000

5. Conclusion

We have presented a study of parallel pipelined architectures of the large point reconfigured FFT algorithm targeting FPGA devices for the implementation of wideband digital receivers. The in-depth exploration of the FFT architectures also can be used in other wideband digital systems. The FFT algorithm adopts two-dimension signal processing, parallelism and pipeline mode, which can be taken into account for a joint assessment. The design includes fixed points unit and variable points unit to realize the configuration of FFT's points that improve the flexibility of system. From our analysis we can conclude that the whole system being miniaturized can alleviates the engineer's work, and the performance just fulfill the requirement of signal processing capabilities and real-time character synchronously.

References

1. Todman.T.J, Constantinides.G.A, Wilton.S.J.E, Mencer.O, and Cheung.P.Y.K, Reconfigurable computing: Architectures and design methods. *IEE Proceedings-Computers and Digital Techniques,* 152(2)(2005), pp. 193-207
2. Grajal.J, Blazquez.R, Lopez-Risueno.G, Sanz.J.M, and Asensio.A, Analysis and characterization of a monobit receiver for electronic warfare. *IEEE Transactions on Aerospace and Electronic Systems,* 39(1)(2003), pp. 244-58.
3. Sanchez M.A, Garrido M, Lopez-Vallejo M and Grajal.J, Implementing FFT-based digital channelized receivers on FPGA platforms. I*EEE Transactions on Aerospace and Electronic Systems,* 44(4)(2008), pp. 1567-85.
4. Chen Erdong, Chen He and Han Yueqiu, The design and implementation of signal processing system based on time/frequency. *Journal of Beijing Institute of Technology,* 25(7)(2005), pp. 636-640.(in Chinese)
5. Kulkarni.A, Jaiswal.V, Girish.M, Suresh.S.P and Revankar.U.K, High speed digital processing for EW receiver applications. *Proceedings of IEEE International Symposium on Microwaves 2008*, pp. 121-125.
6. Chi Dong-ming, Bi Min-li and Qian Chen-cong. The design and realization of a programmable digital EW receiver based on FPGA. Journal of China Academy of Electronics and Information Technology, 2(3)(2007), pp. 288-292.(Chinese)
7. Sanchez.M.A, Garrido.M, Vallejo.M.L and Lopez-Barrio.C, Automated design space exploration of FPGA-based FFT architectures based on area and power estimation. *5th IEEE International Conference On Field Programmable Technology, FPT 2006*, pp. 127-134.
8. Deng.Lanping, Sobti.Kanwaldeep, Zhang.Yuanrui and Chakrabarti.Chaitali, Accurate area, time and power models for FPGA-based implementations. *Journal of Signal Processing Systems,* 63(1)(2011), pp. 39-50.
9. Yun-Nan.C and Parhi.K, An efficient pipelined FFT architecture. *IEEE Transactions on Circuits and Systems II,* 50(6)(2003), pp. 322-325.
10. Sansaloni.T, Pérez-Pascual.A and Valls.J, Area-efficient FPGA-based FFT processor. *Electronics Letters*, 39(19)(2003), pp. 1369-1370.
11. Xiujie Qu, He Chen, Yubin Zhang, Yingtao Ding, Design of DDS based on Hybird-CORDIC Architecture, *International Journal of Computational Intelligence Systems* , 4(4)(2011), pp. 306 -313
12. Wuhong Wang, Hongwei Guo, Katsushi Ikeuchi, Heiner Bubb, Numerical simulation and analysis procedure for digital driving dependability in intelligent transport system, *KSCE Journal of Civil Engineering,* 15(5) (2011), pp. 891-898.
13. Wuhong Wang, Yan Mao, Jin Jing, Xiao Wang, Hongwei Guo, Xuemei Ren, Ikeuchi Katsushi, Driver's various information process and multi-ruled decision-making

mechanism: a fundamental of intelligent driving shaping model, *International Journal of Computational Intelligence Systems*, 4(3)(2011), pp. 297-305.