Design and FPGA implementation of time-frequency transforming for stretch processing

Jian Tan\textsuperscript{a)}, Biyang Wen\textsuperscript{b)}, Yingwei Tian, Ke Li, Jing Yang, and Mao Tian
School of Electronic Information, Wuhan University, Wuhan, 430072, China
\textsuperscript{a)} tanj1989@whu.edu.cn
\textsuperscript{b)} rspl@whu.edu.cn

Abstract: Stretch processing is an efficient method for pulse compression in chirp radar. A FPGA implementation of time-frequency transforming for stretch is presented. The key point is to calculate the spectrum points of interest using the segmental Time-Division Multiplexing (TDM) Discrete Fourier Transform (DFT), and the CORDIC algorithm is used to compute the complex multiplication. This approach shortens time and reduces memory size relative to FFT.

Keywords: stretch processing, pulse compression, time-frequency transforming, TDM-DFT, CORDIC

Classification: Electron devices, circuits, and systems

References

1 Introduction

Pulse compression achieves the high resolution and long range in radar systems [1]. The technique of “stretch processing”, which is a form of matched filtering [2], is applied in the pulse compression of LFM signal. The stretch processing consists of mixing and time-frequency transforming [3, 4].

Usually we use the Fast Fourier Transform (FFT) to achieve the time-frequency transforming in the stretch processing [5]. Considering the situation that the radars receives returns from a few close targets, the constant frequency tones occupy a narrow bandwidth in frequency domain. So only a small part of frequency point is useful after the FFT. Shen put forward a demodulation method to extract the spectrum segment of interest [6]. Compared to the global FFT, the method decreases the computational complexity. But it still needs large memory for raw data and latency for calculation.

This paper provides a new method to overcome the prior shortcomings. The basic idea is segmental DFT which only calculates a few frequency points. TDM is utilized to accomplish the multiplication and addition in the processing of DFT [7]. The coordinate rotation digital computer (CORDIC) algorithm has been used to compute complex multiplications in FFT [8]. Here we use it in DFT for the elimination of multipliers and saving area further. The proposed approach is applied in High-Frequency Surface Wave Radar (HFSWR) and enhances the performances of radar’s hardware system.

The rest of paper is as follows. Section 2 introduces the stretch processing in chirp radar. The TDM-DFT processing based on CORDIC is described in Section 3. The hardware implementation and the results are given in Section 4. At last the conclusion is presented in Section 5.

2 Stretch processing in chirp radar

Stretch processing consists of two steps as following: 1) the returns are mixed with a replica of the transmitted waveform. 2) the time-frequency transforming is used to extract the tones that are proportional to target range. Nowadays the stretch processing can be achieved in digital domain with the development of digital devices [4]. Here we describe it in digital domain.

Assume the sample interval $\Delta T$, then the digital echo of $n$th sampling point in chirp radar can be given by

$$s_r[n] = A \cos[2\pi f_c(n\Delta T - \tau_0) - \pi \alpha(n\Delta T - \tau_0)^2]$$  \hspace{1cm} (1)

Where $A$ is an amplitude constant and proportional to target RCS, antenna gain, and range attenuation, $f_c$ is the carrier frequency, $\alpha$ is the sweep rate, $\tau_0$ represents the delay of the return signal.
Orthogonal transformation is used in the mixing processing to get the Inphase (I) and Quadrature (Q) component \[4, 9\]. So the complex mixing is as

\[ s_m[n] = s_r(n) \exp[-j(2\pi f_r n \Delta T - \pi \alpha (n \Delta T)^2)] \]  

After filtering the high frequency component in (2) and ignoring the amplitude, we obtain a constant frequency tone as following

\[ s_m[n] = \exp[j(2\pi \alpha \tau_0 \Delta T n - 2\pi f_r \tau_0 - \pi \alpha \tau_0^2)] \]  

The equation (3) shows \( s_m[n] \) is a single-frequency signal. Its frequency \( \Delta f = \alpha \tau_0 \) represents the target’s range information and can be obtained when \( s_m[n] \) is transformed from time domain to frequency domain. So Fourier Transform (FT) is applied in stretch processing.

### 3 The TDM-DFT processing based on CORDIC

#### 3.1 TDM Discrete Fourier Transform

As discussed above, the range information can be extracted from time-frequency transform of \( s_m[n] \). The Fourier Transform of \( s_m[n] \) is given by

\[ S[k] = FT\{s_m[n]\} = \sum_{n=0}^{N-1} s_m[n] W_N^k, \quad 0 \leq k \leq N - 1 \]  

Where \( N \) is the length of \( s_m[n] \) and we define the twiddle factor \( W_N^k = e^{-j2\pi kn/N} \).

The frequency resolution of FT result is \( \frac{1}{N\Delta T} \). We assume that the maximum frequency corresponding to the longest detecting range is \( \Delta f_{\text{max}} \), in HFSWR system it locates at \( M = N\Delta T \Delta f_{\text{max}} \ll N \). So only the frequency points \( 0 \leq k \leq M \) are useful. The segmental DFT is taken in the real-time processing. It is shown in (4) that DFT comprises of sine generator, multipliers and addition units. The multiplication with twiddle factors can be done before the next data comes in data flaw. We will call such approach TDM-DFT whose structure is shown in Fig. 1.

![Fig. 1. Structure of TDM-DFT](image)

Now the details of TDM-DFT processing are described in following steps.

1. When \( s_m[n] \) is coming, the sine generator creates twiddle factors \( W_N^k \), \( 0 \leq k \leq M \).
2. Multiply \( s_m[n] \) by the \( M \) twiddle factors, respectively.
3. The results in step 2 are added with the prior ones, respectively.
4. The next data \( s_m[n+1] \) is coming, and the same processing from step 1 will continue.

So the DFT processing can be finished as soon as the data flaw is over. In each stage all \( M \) multiplications and additions have to be finished during the sample interval \( \Delta T \). Different from FFT, the number of input is \( N \) and the output is \( M \) in the segmental DFT. The implementation doesn’t require large ROM for data storage but only a small memory size for the accumulations. Complex multipliers are essential as well as FFT. CORDIC algorithm is used to take place of generation of twiddle factor and multiplication for low cost and area.

### 3.2 CORDIC in place of twiddle factor

There are two modes of operation rotation and vectoring in CORDIC algorithm, and rotation mode provides an efficient method of computing functions by rotating a vector [8]. We use the rotation mode to generate sine function and multiply. Consider the equation (4) again:

\[
 s_m[n]W_{kn}^N = (s_{mx}[n] + js_{my}[n])e^{-j2\pi kn/N} \tag{5}
\]

Where \( s_{mx} \) and \( s_{my} \) is the real part and imaginary part of \( s_m \), respectively. The equation (5) means that vector \( s_m \) is rotated by an angle \( 2\pi kn/N \) on the complex plane as shown in Fig. 2.

The basic idea of CORDIC is breaking down the angle of rotation into a sequence of elementary pre-defined angles, \( \alpha_i = \arctan(2^{-i}) \), so that \( \tan \alpha_i = 2^{-i} \) could be implemented in hardware by shifting through \( i \) bit locations. So the rotation through the angle \( \theta \) is performed by a certain number of microrotations, where

\[
 \theta = \sum_{i=0}^{i-1} \sigma_i \alpha_i, \quad \text{and} \quad \sigma_i = \pm 1 \tag{6}
\]

In addition, the scale factor \( K = \prod_{i=0}^{n} 1/\sqrt{(1 + 2^{-2i})} \) [10] can be ignored because only the relative values are considered in radar system.
Note that the three input ports \((x_{in}, y_{in} \text{ and } \theta)\) are corresponding to the real part, the imaginary part of \(s_m\) and \(-2\pi kn/N\) in a CORDIC module. In our method one CORDIC module is used by time division multiplexing for saving area.

4 The FPGA implementation and results

4.1 The FPGA implementation of TDM-DFT

FPGA XC6SLX150 (XILINX Spartan 6) has been adopted to implement TDM-DFT which is shown in Fig. 3. It consists of phase accumulator, CORDIC module and product accumulator. The pipelined CORDIC architecture [11] is used for shortening time. The HFSWR was designed with carrier frequency of 8 MHz and range resolution of 5 Km, so setting the number of observing range cell \(M = 128\) is enough, which is correspondings to 640 Km. The length of data after mixing and filtering equals \(2^{19}\), with the sampling rate of \(1/\Delta T = 1024\) KHz. A higher clock frequency 162.84 MHz is needed to utilize the one CORDIC module.

![Fig. 3.](image)

The FPGA implementation of TDM-DFT based on CORDIC, ① phase accumulator; ② product accumulator. Where \(s_{mx}\) and \(s_{my}\) are the real and imaginary part of \(s_m\), respectively, \(S_x\) and \(S_y\) are the real and imaginary part of \(S\), respectively.

4.2 The FPGA implementation test and field experiment results

The design and FPGA implementation of the TDM-DFT has been applied in HFSWR system which was developed by Wuhan University [12]. A closed-loop test was employed to check the method via applying a delayed return signal to the receiver. The sweep interval of LFM signal is 512 ms and the return delay is 966 us. The input amplitude of RF was \(-120\) dBm. The result is presented in Fig. 4(a).

A result of stretch processing measured by the HFSWR system on the coast of the Eastern China Sea is represented in Fig. 4(b). The ocean clutter and target echo occupy the dotted line field. It spreads from 0 range cell to 40 range cell. The power decreases with the detecting range increasing which can be explained by radar equation.
4.3 Discussion

In stretch processing the time-frequency transforming is necessary. Compared with FFT and segmental FFT, our approach makes use of the time between sampling data to calculate. FFT and segmental FFT have to wait for the all data’s arriving, their computing complexity are $N \log_2(N)$ and $N \times (1 + \log_2(M))$ [6], respectively. On the contrary TDM-DFT can be finished after $M$ multiplications when the last data are transmitted to the time frequency transforming module. Normally the number of FFT points should be $2^Z$ ($Z$ is positive integer), but DFT method can adapt to any number.

A higher clock is needed in our method because of the TDM. The less $M$ is, then the better this approach works out. With $M$ increasing, we should add a large RAM for twiddle factor and high frequency clock. So it may only be suitable in the situation where the input length of data is much less than the output length.

5 Conclusion

This paper described a time-frequency transforming method in FPGA system and the method usually can be used in the stretch processing. It mainly takes advantage of the TDM-DFT to decrease the storage and shorten the latency. The use of CORDIC algorithm in DFT provides a simpler way of sine generator and complex multiplications. This method has been applied in HFSWR system and the experiment result has shown its practicability.

Acknowledgment

This work is supported by the National Special Program for Key Scientific Instrument and Equipment Development of China under grant 2013YQ160793 and public science and technology research funds projects of ocean under grant 201205032-3. The authors wish to express their gratitude to the editor and the anonymous reviewers.