micromachines

MDPI

Article

An Improved Blind Zone Channelization Structure and Rapid Implementation Method

Ziliang Jia and Hongxia Liu *

Key Laboratory for Wide Band Gap Semiconductor Materials and Devices of Education Ministry, School of Microelectronics, Xidian University, Xi'an 710071, China; 20111223127@stu.xidian.edu.cn
* Correspondence: hxliu@mail.xidian.edu.cn; Tel.: +86-130-8756-8718

Abstract: The paper proposes an enhanced design for broadband digital receivers that aims to improve signal capture probability, real-time performance, and the hardware development cycle. To overcome the issue of false signals in the blind zone channelization structure, this paper introduces an improved joint-decision channelization structure that reduces channel ambiguity during signal reception. Xilinx's high-level synthesis (HLS) tools are used for accelerated algorithm implementation, and techniques such as pipelining and loop parallelization are employed to reduce system latency. The entire system is implemented on FPGA. The simulation results demonstrate that the proposed solution effectively eliminates channel ambiguity, improves algorithm implementation speed, and meets the design requirements.

Keywords: digital channelization; polyphase filtering; joint decision; high level synthesis

1. Introduction

Since the beginning of the 21st century, with the rapid development of modern electronic technology, electronic information warfare has become a major form of warfare between nations. In electronic warfare equipment, more and more advanced technologies are being applied in the fields of jamming, anti-jamming, reconnaissance, and counter-reconnaissance, making the electromagnetic environment increasingly complex [1,2]. At the same time, the radio signals generated by these devices also vary in signal energy and carrier frequency size. To adapt to such a complex environment, electronic warfare receivers need to have a wide monitoring frequency band range, real-time signal processing, a large dynamic range, and the ability to process time–domain overlapping signals [3–5].

Traditional electronic warfare receivers mostly use analog technology design, including crystal video receivers, compression receivers, superheterodyne receivers, Bragg receivers, instantaneous frequency measurement (IFM) receivers [6], and channelized receivers. Crystal video receivers have the advantage of a simple structure but can only be used for signal detection and cannot measure signal frequency. Compression receivers use the Fourier transform to compress input signals of different frequencies into time–domain pulse signals, which has good data processing performance, but signal compression may lead to system detection errors. Superheterodyne receivers typically use multi-stage mixing to convert the input RF (radio frequency) signal to intermediate frequency (IF) for processing at a lower frequency, with a good reception dynamic range, but requiring the introduction of filters to eliminate local oscillator leakage, increasing the receiver cost and complexity. Bragg receivers convert input signals into optical signals through an optical Bragg cell [7], and use a small amount of hardware resources to realize a large number of channels, but have a small reception dynamic range and low sensitivity. IFM receivers mainly use the non-linear characteristics of crystal detectors to measure the frequency of input signals, which can cover a larger monitoring frequency range, with high frequency measurement accuracy, but poor frequency measurement accuracy for time–domain overlapping signals, making it difficult to meet the actual electronic warfare requirements.

check for updates

Citation: Jia, Z.; Liu, H. An Improved Blind Zone Channelization Structure and Rapid Implementation Method. *Micromachines* **2023**, *14*, 1091. https://doi.org/10.3390/mi14051091

Academic Editor: José de Jesús Rangel Magdaleno

Received: 21 April 2023
Revised: 11 May 2023
Accepted: 19 May 2023
Published: 22 May 2023

Channelized receivers divide the working frequency band into multiple channels and, after channelization processing, signals of different frequencies appear in different channels with a large monitoring bandwidth and reception sensitivity, capable of processing time–domain overlapping signals; these receivers can meet the actual electronic warfare requirements and thus have important research value.

Due to the limitations of analog devices, traditional receivers have a good performance but are heavy and bulky, with poor flexibility [8]. With the rapid development of electronic information technology, more and more digital signal processing methods have been applied in the field of digital receivers. At the same time, with the continuous breakthrough of high-speed ADC (analog-to-digital converter) sampling technology, its sampling rate has been increased from MHz to GHz, enabling RF signals to be directly sampled, which also puts higher demands on backend data processing [9,10].

The ability to achieve full probability reception of signals across the entire frequency band is the most important performance metric for wideband digital receivers. A simple and effective method is to construct an overlapped digital channelization model, where the channel parts overlap and the processed signals are seamlessly joined in the frequency domain [11,12]. Although this method can achieve blind-spot-free signal reception, there is some wasting of the frequency spectrum due to the overlapping of the passbands. Another method is to use interpolated half-band filters to decompose the input signal into two complementary and non-overlapping channel signals in the frequency spectrum, and then filter each channel signal with an analysis filter bank to achieve channelization [13]. Although this method can correctly receive wideband channelized signals, its complex structure increases the difficulty of hardware implementation. Based on the limitations of the above methods, this article uses a 50% overlapping filter bank partitioning method [14] to construct filters and proposes an improved decision algorithm to eliminate the introduced ambiguity between channels. This method can avoid wasting spectral resources and is more suitable for hardware implementation. After introducing the decision algorithm proposed in this paper, it can effectively improve the signal acquisition capability of the receiver and improve the resolution of the output signal.

The limited application of broadband digital receivers is partially due to the mismatch between the speed of hardware development and theoretical development. With the rapid development of integrated circuit technology, digital receivers based on field-programmable gate arrays (FPGAs) have a good performance and can meet practical needs [15]. FPGAs have a relatively smaller size and weight compared with clusters of computers, multi-core processors, and many-core processors, etc. [16].

In the hardware implementation of the digital channelization receiver, if the traditional FPGA development process is used, the debugging and development cycle of the entire system will also increase when the data volume is large. Refs. [17,18] have both used the traditional FPGA development process, and when the number of channels is large, the delay of each part will also increase, leading to unreasonable timing in the system. In this article, a new development tool, the HLS development platform, is used in combination. To address problems in traditional FPGA development processes, such as long development cycles and tedious debugging processes, this tool uses the C/C++ high-level programming language to design and implement algorithms, and synthesizes the designed program into Verilog language to complete the transformation to the RTL level [19]. In the synthesis process, optimization instructions such as pipeline and loop parallelism can be added to reduce system delay. By combining the HLS platform with the traditional development process, the hardware implementation efficiency and design flexibility of the entire receiver system can be greatly improved.

This paper presents an improved method to address the issues of false signals in digital receivers. The proposed method transforms the process of eliminating false signals into extracting amplitude-frequency information from the output signals. This not only improves the discrimination of useful signals, but also simplifies the implementation process of FPGAs. Additionally, the paper employs a hybrid design approach by combining

the HLS design approach [19] with traditional methods to accelerate the implementation process of front-end algorithms, resulting in increased hardware implementation efficiency and design flexibility for the entire receiver system.

2. Polyphase Filtering Digital Channelization Model

2.1. The Original Structure

For the digital channelization filter bank, the input signal s(n) is down-converted and then evenly divided into K subbands by K filters. After D-fold decimation, the output results are obtained, and this process satisfies:

$$K = F \cdot D \tag{1}$$

When $F = 1$, it becomes a critical decimation process, and the original structure of the channelized receiver is shown in Figure 1.

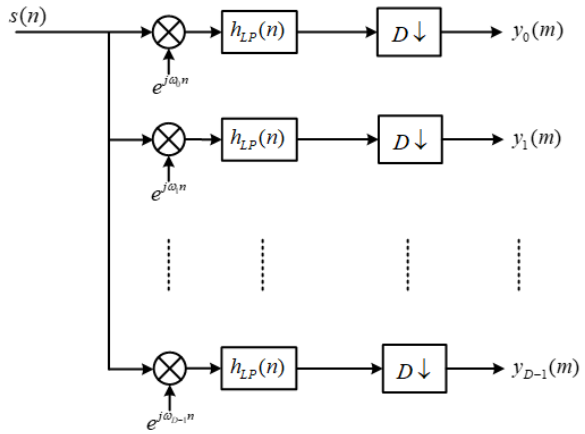


Figure 1. Original structure of digital channelized reception.

The input signal is first modulated and then shifted to baseband. Afterward, it undergoes low-pass filtering and decimation before being fed into subsequent modules for further processing. However, at this stage, the filtering operation occurs before decimation, leading to redundant computations during signal processing. Moreover, convolution operations must be completed within a single sampling period. To address these issues, an alternative approach is proposed, which involves swapping the positions of filtering and decimation operations. This way, the computational requirements of various modules in the system can be met more efficiently.

2.2. Subchannelization

Multi-rate signal processing with the polyphase filtering technique plays a crucial role in reducing the complexity of signal rate conversion and lowering system design complexity. The key process in this technique is subchannelization, where the target monitoring frequency band is divided according to certain rules, and different partitioning methods correspond to different system complexities, computational loads, and resource consumption. In this design, the filter group adopts a 50% overlapping partitioning method [20], as shown in Figure 2, which concatenates the passbands of adjacent subchannels, eliminating any blind spots in the entire monitoring bandwidth and achieving the full probability of interception.

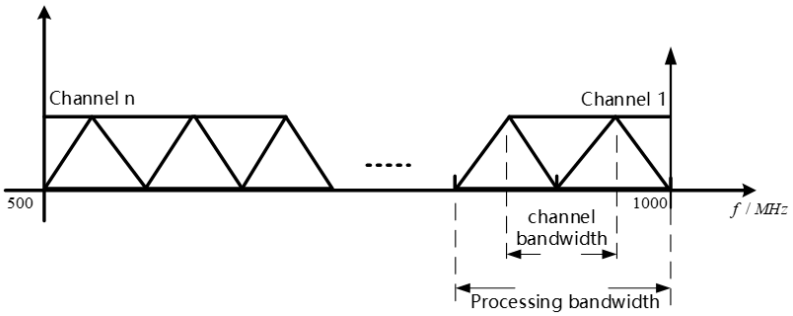


Figure 2. Adjacent channel with 50% overlapping uniform channel partitioning.

2.3. Improved Non-Blind Channelization Structure

The DFT-based polyphase channelization algorithm is developed based on the channelization structure of the low-pass filter bank. Given the input signal $s(n)$, the output $f_k(m)$ of the kth subchannel is defined as:

$$\begin{aligned} y_k(m) &= \left\{ [s(n)e^{j\omega_k n}] * h(n) \right\}|_{n=mD} \\ &= \left\{ \sum_{i=-\infty}^{+\infty} s(n-i)e^{j\omega_k(n-i)} \cdot h(i) \right\}\Bigg|_{n=mD} \\ &= \sum_{i=-\infty}^{+\infty} s(mD-i)e^{j\omega_k(mD-i)} \cdot h(i) \\ &= \sum_{p=0}^{D-1} \sum_{i=-\infty}^{+\infty} s(mD-iD-p)e^{j\omega_k(mD-iD-p)} \cdot h(iD+p) \end{aligned} \tag{2}$$

Now, let $s_p(m) = s(mD-p)$ and $h_p(m) = h(mD+p)$. Then, we can further simplify Equation (2) as follows:

$$y_k(m) = \sum_{p=0}^{D-1} \left[\sum_{i=-\infty}^{+\infty} s_p(m-i)e^{j\omega_k(m-i)D} \cdot h_p(i) \right] e^{-j\omega_k p} \tag{3}$$

Convert part of the relationship in Equation (3) to:

$$x_p(m) = \sum_{i=-\infty}^{+\infty} s_p(m-i)e^{j\omega_k(m-i)D} \cdot h_p(i) = \left[s_p(m)e^{j\omega_k mD} \right] * h_p(m) \tag{4}$$

The center frequency of each subchannel in the filter bank under the even-indexed arrangement is:

$$\omega_k = \frac{2\pi k}{D} \tag{5}$$

Substituting Equation (5) into Equation (4), we obtain:

$$x_p(m) = \left[s_p(m)e^{jm\frac{2\pi k}{D} \cdot D} \right] * h_p(m) = s_p(m) * h_p(m) \tag{6}$$

Substituting Equation (6) into Equation (3), we obtain:

$$y_k(m) = \sum_{p=0}^{D-1} x_p(m) \cdot e^{-j\frac{2\pi}{D}kp} = DFT\left[s_p(m) * h_p(m) \right] \tag{7}$$

In the above equation, DFT represents discrete Fourier transform, which can be replaced by fast Fourier transform (FFT) in hardware implementation to reduce hardware resource consumption and improve system operation speed. The improved structure is shown in the following Figure 3. This structure eliminates the multiplier factors used in traditional structures, resulting in improvements in the data path and resource utilization.

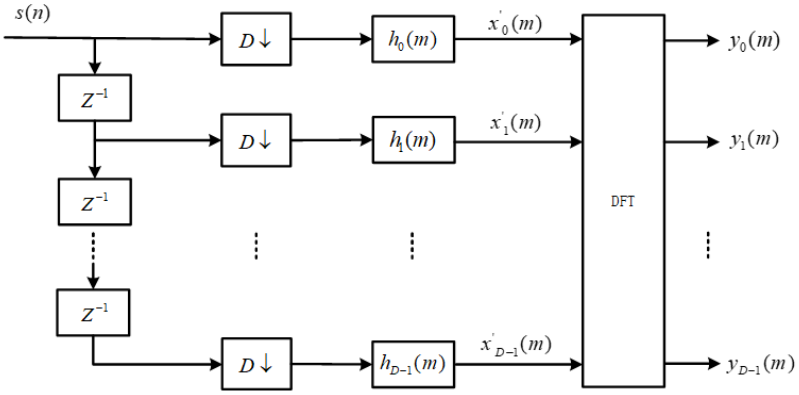


Figure 3. Efficient Digital Channelizer Receiver Model for Complex Signals.

2.4. Improved Channel Decision Module

The proposed joint-decision process for eliminating channel ambiguity in the channelized receiver involves "Instantaneous Feature Extraction + Auto-Correlation Threshold Amplitude Detection + Phase Differential Instantaneous Frequency Measurement". In the instantaneous feature extraction step, the coordinate rotation digital computer (CORDIC) algorithm is used to extract the instantaneous amplitude and phase information of the channelized output signal in vector mode. Traditional amplitude detection methods compare the signal amplitude with a fixed threshold value to determine the presence of signals in the channel [21]. In this study, the auto-correlation amplitude detection algorithm is used, which compares the signal amplitude information with dynamically changing threshold values that depend on the amplitudes of multiple points, resulting in more accurate detection results [22]. The phase differential instantaneous frequency measurement utilizes the extracted instantaneous phase information to measure the instantaneous frequency of the signal and, based on the frequency measurement result and the corresponding relationship with the channel center frequency, the correct channel output is determined, eliminating channel ambiguity. Figure 4 illustrates the joint-decision process, which ultimately yields accurate channel output results.

The CORDIC (coordinate rotation digital compute) algorithm calculates the amplitude and phase information of the output signal through iterative rotations [23]. The solving process is shown in Equation (8):

$$\begin{aligned} \alpha_k(n) &= \sqrt{I_k{}^2(n) + Q_k{}^2(n)} \\ \varphi_k(n) &= \arctan\left[\frac{Q_k(n)}{I_k(n)}\right] \\ f_k(n) &= \frac{f_s}{2\pi}[\varphi_k(n) - \varphi_k(n-1)] \end{aligned} \tag{8}$$

$I_k(n)$ and $Q_k(n)$ represent the real and imaginary parts of the output of the kth subchannel, and $\alpha_k(n)$, $\varphi_k(n)$, and $f_k(n)$ represent the instantaneous amplitude, phase, and frequency of the signal, respectively. The threshold value for adaptive detection is set based on the amplitude information of the output signal, and it is determined according to Equation (9):

$$V_T[n] = \beta \cdot \mu + \gamma = \beta \cdot \frac{1}{D} \cdot \sum_{k=1}^{D} A_k[n] + \gamma \tag{9}$$

where β represents the threshold coefficient, $A_k[n]$ represents the amplitude of the output signal from the kth subchannel, D represents the extraction factor under critical extraction state, $V_T[n]$ represents the adaptive detection threshold, and γ represents the noise floor introduced by the receiver. The amplitude value of the channelized output signal varies dynamically with the input signal, so the threshold values determined according to Equation (9) are also dynamic.

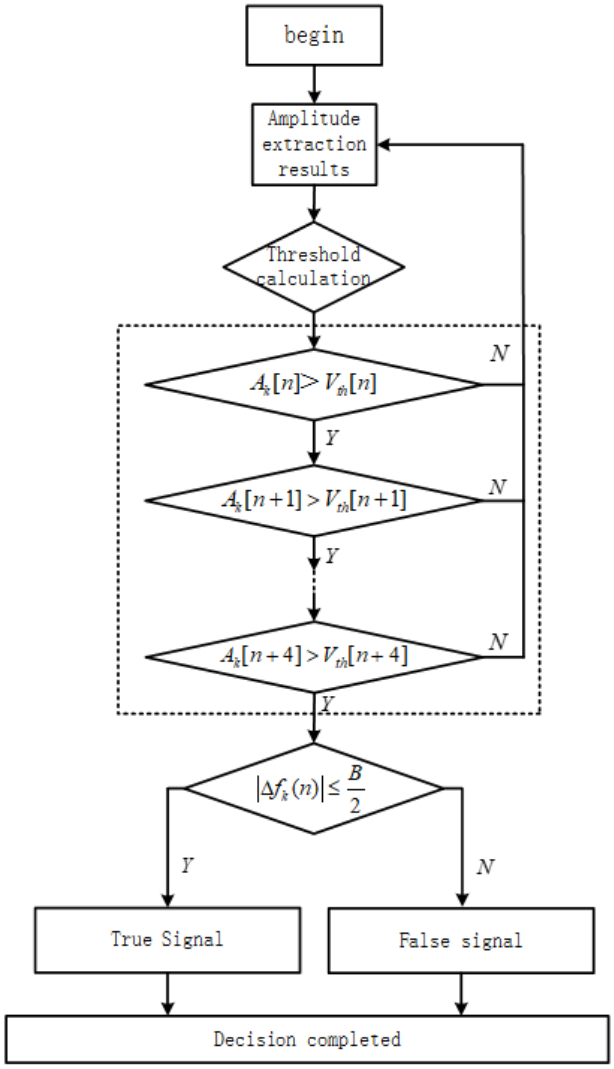


Figure 4. Joint-decision process.

The instantaneous frequency of the output signal can be obtained using the phase difference instantaneous frequency measurement method, where the relationship between instantaneous frequency and instantaneous phase is given by Equation (10):

$$f_k(n) = \frac{\varphi_k(n) - \varphi_k(n-1)}{2\pi T} \tag{10}$$

T represents the sampling period of the signal. The frequency of the output signal can be obtained using the four-point phase difference averaging method to reduce measurement errors caused by noise, as the phase difference instantaneous frequency measurement method is sensitive to noise. In this design, the frequency of the output signal is given by the result obtained from the four-point phase difference averaging method.

$$f_k(n) = \frac{\sum_{i=0}^{3} \Delta\varphi(n)}{8\pi T} \tag{11}$$

The entire decision-making process is as follows:

1. Set the detection threshold $V_T[n]$.
2. Use the signal amplitude calculated from the CORDIC module as input. To reduce the impact of the signal-to-noise ratio, consider a signal to be present in the channel if the

amplitude values $A_k[n]$ are greater than the detection threshold for five consecutive times. Then, proceed to the channel decision-making part.

3. Finally, perform a phase-difference-based instantaneous frequency estimation for the channel. The frequency deviation value $\Delta f_k(n)$ represents the frequency deviation of the signal from the center of the channel. If $|\Delta f_k(n)|$ is less than half of the bandwidth, B, of the channel, it is considered that the signal is within the current channel, otherwise it is considered a false signal.

3. Simulation and Implementation of Improved Structure

3.1. Receiver System Simulation

Simulation Conditions: In MATLAB, the input signal is set as the sum of four signals from Table 1. The input signal-to-noise ratio (SNR) is set to 10 dB, with a total of 4096 sampling points. The number of digital channelized channels is 16. The prototype filter is set to have an order of 192 and a stopband attenuation of 60 dB. The frequency range for each sub-channel is shown in Table 2.

Table 1. Information for setting up the overlapped input signals for simulation.

Input Signals	Frequency (MHz)	Amplitude (V)
Signal 1-sin	935	1.5
Signal 2-chirp	870–890	2
Signal 3-sin	720	2
Signal 4-chirp	590–600	1.5

Table 2. Subchannel frequency range distribution table.

Channel Number	Channel Center Frequency (MHz)	Range of Subchannel Frequency (MHz)
1	1000	1000–984.375
2	968.75	984.375~953.125
3	937.5	953.125~921.875
4	906.25	921.875~890.625
5	875	890.625~859.375
6	843.75	859.375~828.125
7	812.5	828.125~796.875
8	781.25	796.875~765.625
9	750	765.625~734.375
10	718.75	734.375~703.125
11	687.5	703.125~671.875
12	656.25	671.875~640.625
13	625	640.625~609.375
14	593.75	609.375~578.125
15	562.5	578.125~546.875
16	531.25	546.875~515.625

The channelized output results are shown in Figure 5. The simulation experiment tested the receiving capability of the channelized receiver for different input signals. The parameters of the four input signals were set in a controlled manner, selecting four typical frequencies to ensure that each signal belonged to a different channel. From Figure 5, it can be observed that the receiver model was able to correctly channelize and receive the superimposed signals from the four different input signals.

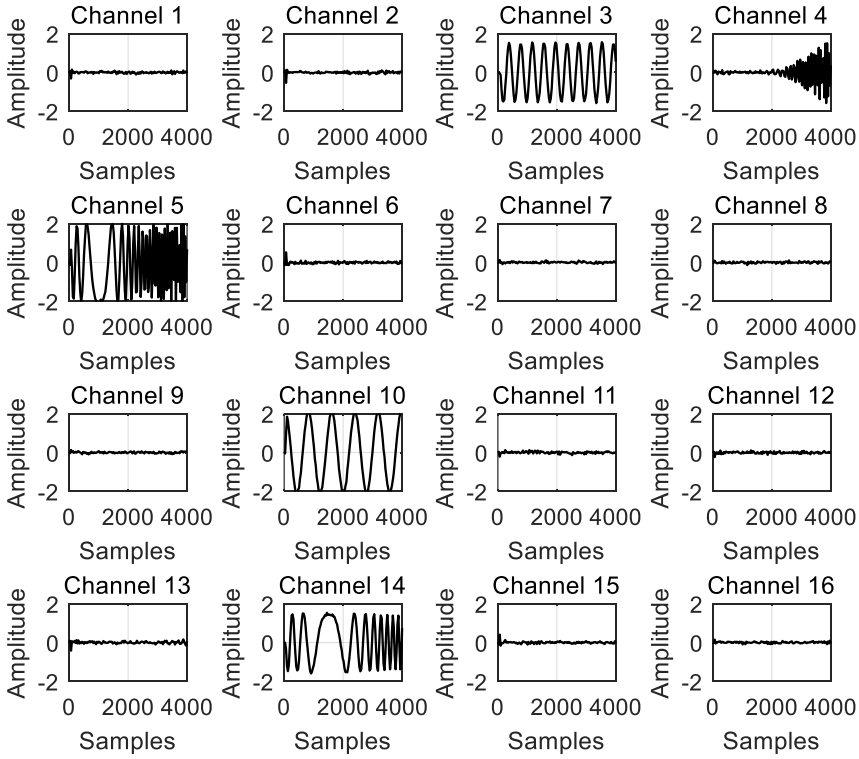


Figure 5. The channelized output results for 16 channels.

For the channels with detected signals, the extracted amplitude, phase, and frequency information are shown in Figures 6 and 7 (taking channel 3 and channel 14 as examples). From left to right and top to bottom in Figure 6, we have the following results:

Amplitude extraction result for channel 3, Phase extraction result for channel 3, Amplitude extraction result for channel 14 and Phase extraction result for channel 14. From left to right in Figure 7, we have the following results: Frequency measurement result for channel 3 and Frequency measurement result for channel 14.

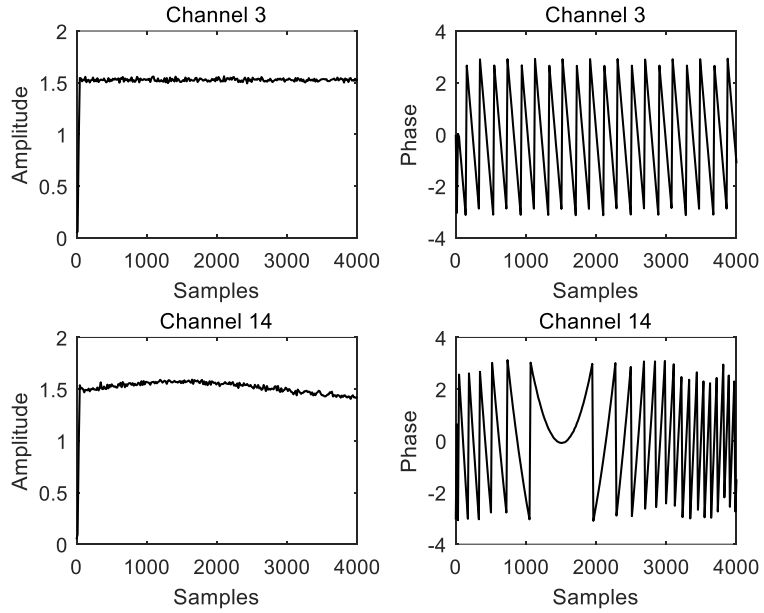


Figure 6. The amplitude and phase extraction results.

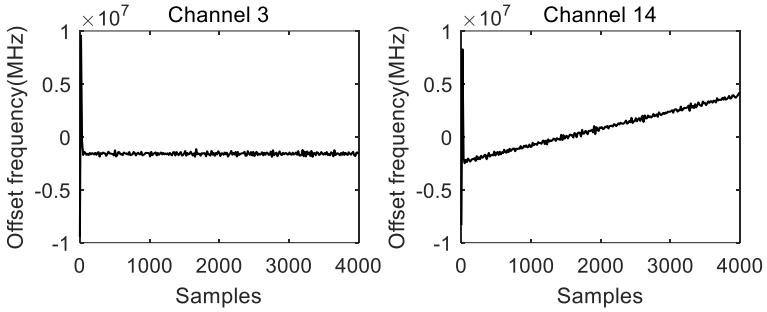


Figure 7. Instantaneous frequency measurement results.

3.2. Hardware Implementation

The hardware implementation of the wideband digital channelized receiver primarily focuses on the large-scale and high-speed digital signal processing. The hardware platform is based on the Zynq UltraScale + RFSoC series chip, specifically the ZU27DR, which includes high-speed ADC and FPGA sections. The improved scheme of the entire receiver system is shown in Figure 8.

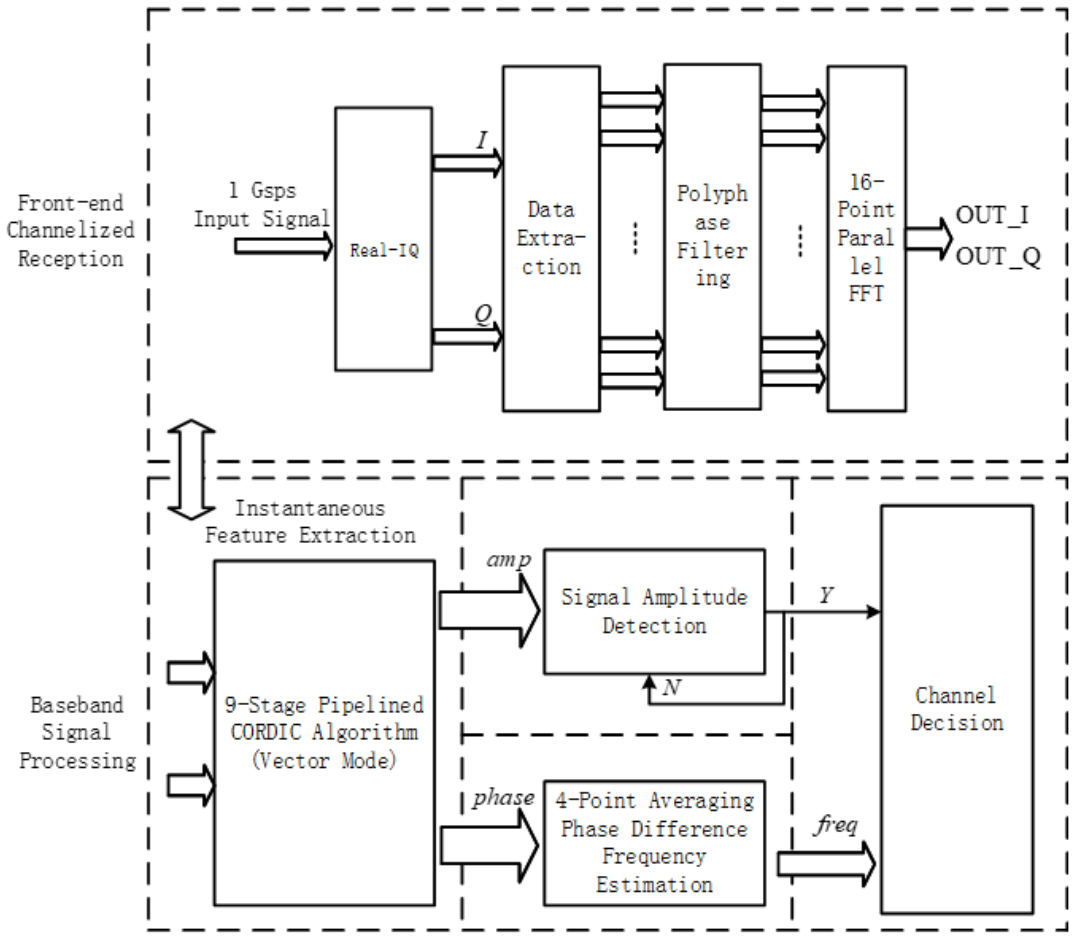


Figure 8. Improved Scheme for Wideband Digital Channelization Receiver.

3.3. Data Extraction and Routing Module

In the HLS platform, when implementing the data extraction and routing module, input data can be stored in a multi-dimensional array based on the storage format of the data stream. During the transformation process, the input data needs to be rearranged by columns for parallel computation. Optimization directives can be added during the C

Synthesis stage to reduce system latency [24]. Since the four loops in the data extraction and routing module have clear boundaries, and the input signal is divided into I and Q paths, there is no dependency between the real and imaginary part computations. Therefore, optimization can be achieved using the "Pipeline + Rewind" approach, as shown in Figure 9, to improve the parallel processing capability of the hardware.

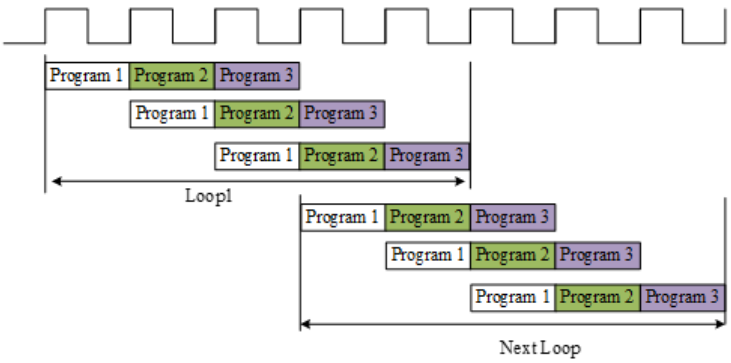


Figure 9. "Pipeline + Rewind" Optimization approaches.

3.4. Polyphase Filtering Module

In HLS, when designing the polyphase filtering module, the prototype filter coefficients need to be stored in an array. The coefficients are stored in an array of size (16, (filter order/16)). The implementation process of polyphase filtering is similar to convolution in the time domain, with the only difference being that FIR filtering requires zero padding at the end of the input sequence and uses a circular right shift to align the first item of the input sequence with the first item of the coefficient. Each right shift requires a multiplication operation, and the multiplication results are stored in registers. The final accumulated result is the output of the FIR filter. The polyphase filtering module consists of 16 parallel channels, and since the structure of each channel is the same except for the coefficients and input data, the implementation process of any sub-channel is shown in Figure 10.

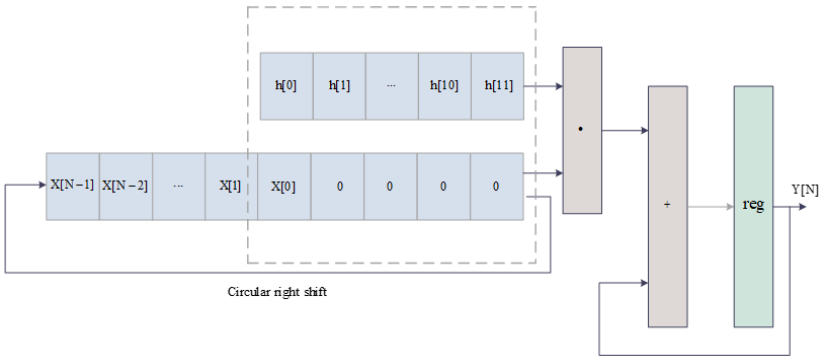


Figure 10. The implementation process of FIR filtering in any sub-channel.

3.5. The Fully Parallel FFT Module

Due to the high throughput requirements of the data output from the multi-phase filtering structure during FFT computation, a parallel FFT module that can perform parallel computation is needed [25]. However, the FFT IP core provided by Xilinx requires at least 16 clock cycles to complete a 16-point DFT computation [26], which does not meet the design requirements. In this paper, a parallel FFT structure is designed using a pipelined design approach, based on the radix-2 decimation in time (DIT) method, to compute the 16-point DFT. The core of the FFT algorithm is the butterfly operation, which divides the 16-point DFT into two 8-point DFTs based on the parity of the input sequence x(n), and further divides the 8-point DFTs into two 4-point DFTs, and so on, until the 16-point DFT is

transformed into multiple 2-point DFT computations [27,28]. The butterfly operation flow of the 16-point parallel FFT module is shown in Figure 11.

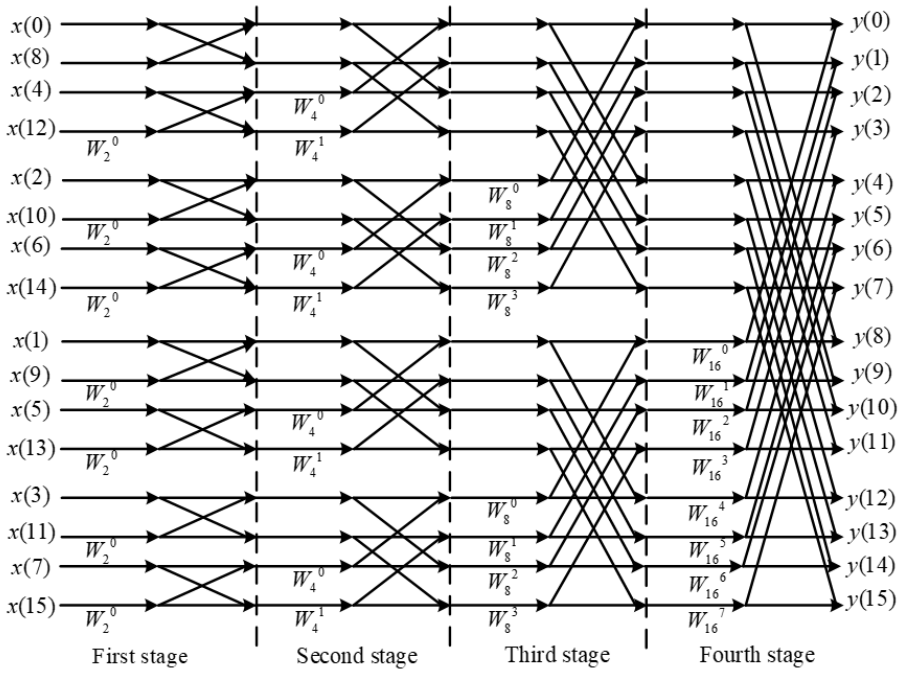


Figure 11. The 16-point FFT parallel processing flow.

3.6. Channel Decision Module

The CORDIC algorithm has three hardware implementation architectures: serial architecture, parallel architecture, and parallel pipeline architecture. These three architectures are based on the same basic structure of processing units, but operate at different shift amounts and storage angles, resulting in different implementation methods. In this paper, the parallel pipeline architecture is chosen for its design, which consumes significantly more hardware resources compared to the serial architecture.

By adding pipeline registers between each iteration, the processing speed of the system can be effectively improved, reducing the critical path length from N processing units in the parallel architecture to 1 processing unit in the pipeline architecture.

After obtaining the phase of the complex signal through the CORDIC module, the instantaneous frequency of the signal is calculated using the phase-difference-based frequency estimation algorithm, as shown in Figure 12.

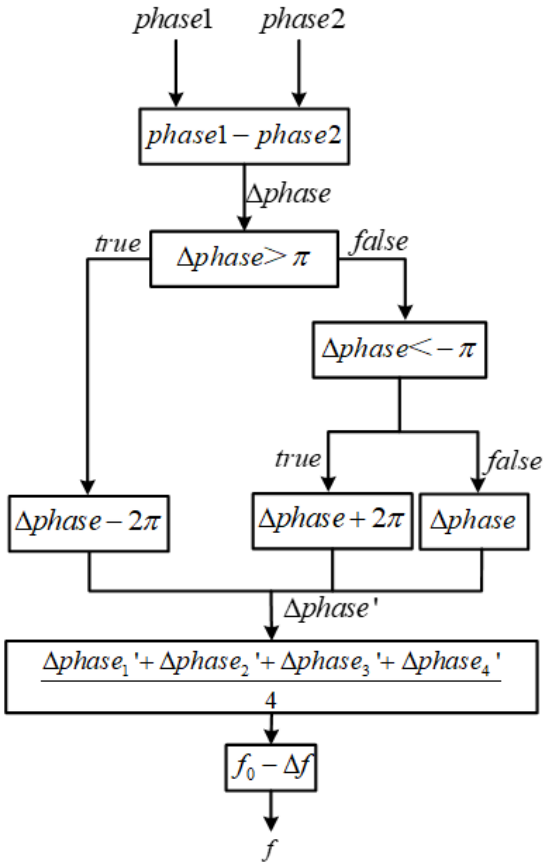


Figure 12. Instantaneous frequency measurement solution process.

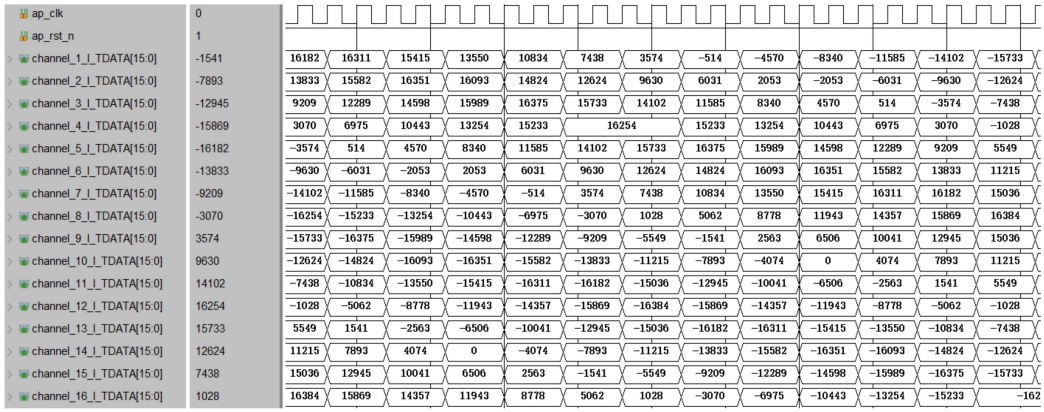
4. Simulation and Analysis

The ChipScope, an integrated logic analyzer provided by Xilinx, was used to observe the signal, as described in [29]. In MATLAB, a sine wave signal with a carrier frequency of 935 MHz was generated and simulated. Since the input test signal is a complex signal, one path was selected for extraction and comparison verification in the real and imaginary parts. Here, the I path was selected for comparison verification. The first 10 columns of the extracted simulation results in MATLAB are shown in Figure 13.

	1	2	3	4	5	6	7	8	9	10
1	16182	16311	15415	13550	10834	7438	3574	-515	-4571	-8341
2	13833	15582	16351	16093	14824	12624	9630	6031	2053	-2054
3	9209	12289	14598	15989	16375	15733	14102	11585	8340	4570
4	3070	6975	10443	13254	15233	16254	16254	15233	13254	10443
5	-3575	514	4570	8340	11585	14102	15733	16375	15989	14598
6	-9631	-6032	-2054	2053	6031	9630	12624	14824	16093	16351
7	-14103	-11586	-8341	-4571	-515	3574	7438	10834	13550	15415
8	-16255	-15234	-13255	-10444	-6976	-3071	1028	5062	8778	11943
9	-15734	-16376	-15990	-14599	-12290	-9210	-5550	-1542	2563	6506
10	-12625	-14825	-16094	-16352	-15583	-13834	-11216	-7894	-4075	0
11	-7439	-10835	-13551	-15416	-16312	-16183	-15037	-12946	-10042	-6507
12	-1029	-5063	-8779	-11944	-14358	-15870	-16384	-15870	-14358	-11944
13	5549	1541	-2564	-6507	-10042	-12946	-15037	-16183	-16312	-15416
14	11215	7893	4074	0	-4075	-7894	-11216	-13834	-15583	-16352
15	15036	12945	10041	6506	2563	-1542	-5550	-9210	-12290	-14599
16	16384	15869	14357	11943	8778	5062	1028	-3071	-6976	-10444

Figure 13. Simulation results of the extracted channel in MATLAB.

The data shown in Figure 13 are obtained after quantization. The timing simulation results of the extraction path module are shown in Figure 14. After comparing with the simulation results in Matlab, it can be confirmed that the function of the module is correct.



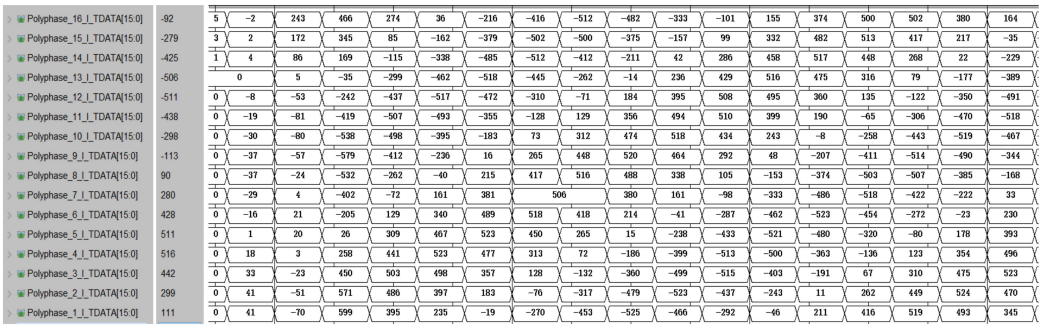
ap_clk	0															
ap_rst_n	1															
channel_1_I_TDATA[15:0]	-1541	16182	16311	15415	13550	10834	7438	3574	−514	−4570	−8340	−11585	−14102	−15733		
channel_2_I_TDATA[15:0]	-7893	13833	15582	16351	16093	14824	12624	9630	6031	2053	−2053	−6031	−9630	−12624		
channel_3_I_TDATA[15:0]	-12945	9209	12289	14598	15989	16375	15733	14102	11585	8340	4570	514	−3574	−7438		
channel_4_I_TDATA[15:0]	-15869	3070	6975	10443	13254	15233	16254		15233	13254	10443	6975	3070	−1028		
channel_5_I_TDATA[15:0]	-16182	−3574	514	4570	8340	11585	14102	15733	16375	15989	14598	12289	9209	5549		
channel_6_I_TDATA[15:0]	-13833	−9630	−6031	−2053	2053	6031	9630	12624	14824	16093	16351	15582	13833	11215		
channel_7_I_TDATA[15:0]	-9209	−14102	−11585	−8340	−4570	−514	3574	7438	10834	13550	15415	16311	16182	15036		
channel_8_I_TDATA[15:0]	-3070	−16254	−15233	−13254	−10443	−6975	−3070	1028	5062	8778	11943	14357	15869	16384		
channel_9_I_TDATA[15:0]	3574	−15733	−16375	−15989	−14598	−12289	−9209	−5549	−1541	2563	6506	10041	12945	15036		
channel_10_I_TDATA[15:0]	9630	−12624	−14824	−16093	−16351	−15582	−13833	−11215	−7893	−4074	0	4074	7893	11215		
channel_11_I_TDATA[15:0]	14102	−7438	−10834	−13550	−15415	−16311	−16182	−15036	−12945	−10041	−6506	−2563	1541	5549		
channel_12_I_TDATA[15:0]	16254	−1028	−5062	−8778	−11943	−14357	−15869	−16384	−15869	−14357	−11943	−8778	−5062	−1028		
channel_13_I_TDATA[15:0]	15733	5549	1541	−2563	−6506	−10041	−12945	−15036	−16182	−16311	−15415	−13550	−10834	−7438		
channel_14_I_TDATA[15:0]	12624	11215	7893	4074	0	−4074	−7893	−11215	−13833	−15582	−16351	−16093	−14824	−12624		
channel_15_I_TDATA[15:0]	7438	15036	12945	10041	6506	2563	−1541	−5549	−9209	−12289	−14598	−15989	−16375	−15733		
channel_16_I_TDATA[15:0]	1028	16384	15869	14357	11943	8778	5062	1028	−3070	−6975	−10443	−13254	−15233	−162		

Figure 14. Time–domain simulation result of the extraction branch module.

The MATLAB simulation result and timing simulation result of the polyphase filter module are shown in Figures 15 and 16, respectively. After comparison, it was found that the simulation results in HLS were correct, thus completing the consistency verification between the multi-phase filter module and the theoretical model.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0	41	-70	599	395	235	-19	-269	-453	-525	-466	-292	-46	211	416	519
2	0	41	-51	571	486	397	183	-76	-317	-479	-523	-437	-243	11	262	449
3	0	33	-23	450	503	498	357	128	-132	-360	-499	-515	-403	-191	67	310
4	0	18	3	258	441	523	477	313	72	-186	-399	-513	-500	-363	-136	123
5	0	1	20	26	309	467	523	450	265	15	-238	-433	-521	-480	-320	-80
6	0	-16	21	-205	129	340	489	518	418	214	-41	-287	-462	-523	-454	-272
7	0	-29	4	-402	-72	161	381	506	506	380	161	-98	-333	-486	-518	-422
8	0	-37	-24	-532	-262	-40	215	417	516	488	338	105	-153	-374	-503	-507
9	0	-37	-57	-579	-412	-236	16	265	448	520	464	292	48	-207	-411	-514
10	0	-30	-80	-538	-498	-395	-183	73	312	474	518	434	243	-8	-258	-443
11	0	-19	-81	-419	-507	-493	-355	-128	129	356	494	510	399	190	-65	-306
12	0	-8	-53	-242	-437	-517	-472	-310	-71	184	395	508	495	360	135	-122
13	0	0	5	-35	-299	-462	-518	-445	-262	-14	236	429	516	475	316	79
14	1	4	86	169	-115	-338	-485	-512	-412	-211	42	286	458	517	448	268
15	3	2	172	345	85	-162	-379	-502	-500	-375	-157	99	332	482	513	417
16	5	-2	243	466	274	36	-216	-416	-512	-482	-333	-101	155	374	500	502

Figure 15. Polyphase filtering results in MATLAB.



Polyphase_16_I_TDATA[15:0]	-92	5	−2	243	466	274	36	−216	−416	−512	−482	−333	−101	155	374	500	502	380	164
Polyphase_15_I_TDATA[15:0]	-279	3	2	172	345	85	−162	−379	−502	−500	−375	−157	99	332	482	513	417	217	−35
Polyphase_14_I_TDATA[15:0]	-425	1	4	86	169	−115	−338	−485	−512	−412	−211	42	286	458	517	448	268	22	−229
Polyphase_13_I_TDATA[15:0]	-506	0	5	−35	−299	−462	−518	−445	−262	−14	236	429	516	475	316	79	−177	−389	
Polyphase_12_I_TDATA[15:0]	-511	0	−8	−53	−242	−437	−517	−472	−310	−71	184	395	508	495	360	135	−122	−350	−491
Polyphase_11_I_TDATA[15:0]	-438	0	−19	−81	−419	−507	−493	−355	−128	129	356	494	510	399	190	−65	−306	−470	−518
Polyphase_10_I_TDATA[15:0]	-298	0	−30	−80	−538	−498	−395	−183	73	312	474	518	434	243	−8	−258	−443	−519	−467
Polyphase_9_I_TDATA[15:0]	-113	0	−37	−57	−579	−412	−236	16	265	448	520	464	292	48	−207	−411	−514	−490	−344
Polyphase_8_I_TDATA[15:0]	90	0	−37	−24	−532	−262	−40	215	417	516	488	338	105	−153	−374	−503	−507	−385	−168
Polyphase_7_I_TDATA[15:0]	280	0	−29	4	−402	−72	161	381	506		380	161	−98	−333	−486	−518	−422	−222	33
Polyphase_6_I_TDATA[15:0]	428	0	−16	21	−205	129	340	489	518	418	214	−41	−287	−462	−523	−454	−272	−23	230
Polyphase_5_I_TDATA[15:0]	511	0	1	20	26	309	467	523	450	265	15	−238	−433	−521	−480	−320	−80	178	393
Polyphase_4_I_TDATA[15:0]	516	0	18	3	258	441	523	477	313	72	−186	−399	−513	−500	−363	−136	123	354	496
Polyphase_3_I_TDATA[15:0]	442	0	33	−23	450	503	498	357	128	−132	−360	−499	−515	−403	−191	67	310	475	523
Polyphase_2_I_TDATA[15:0]	299	0	41	−51	571	486	397	183	−76	−317	−479	−523	−437	−243	11	262	449	524	470
Polyphase_1_I_TDATA[15:0]	111	0	41	−70	599	395	235	−19	−270	−453	−525	−466	−292	−46	211	416	519	493	345

Figure 16. The timing simulation results of the polyphase filtering module.

The final channelized output result is shown in the Figure 17, where PFFT_OUT_3I is the real component of the third channel output, and PFFT_OUT_3Q is the imaginary component of the third channel output. It can be seen that there is an IQ component output in the third channel, proving that the calculation result of this module in FPGA is correct.

In addition, a chirp signal with a center frequency of 875 MHz and a bandwidth of 10 MHz was generated for the simulation of instantaneous feature extraction. The simulation results, depicted in Figures 18 and 19, demonstrate the improved channel decision module's ability to accurately extract amplitude, phase, and frequency information from the received signal. By comparing the extracted signal information with the sub-channel center frequencies defined in the receiver, false signals can be eliminated, resulting in a blind-zone-free reception in the receiver.

Figure 17. Time–domain simulation result of channelized output.

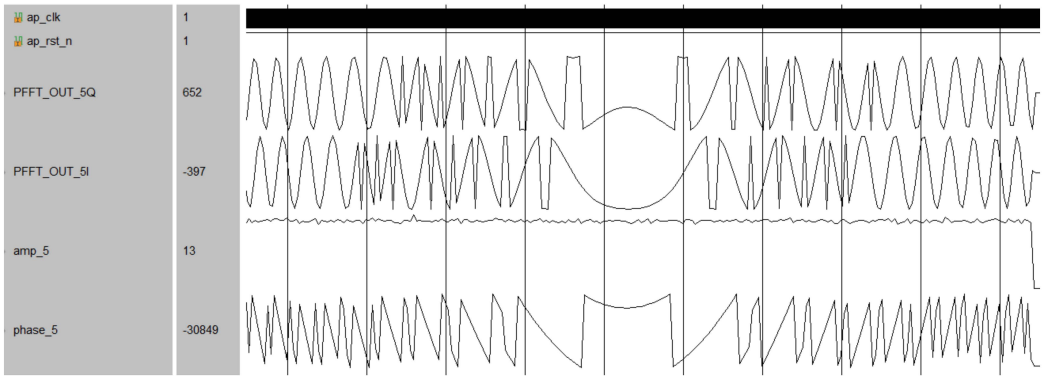


Figure 18. The results of amplitude and phase extraction.

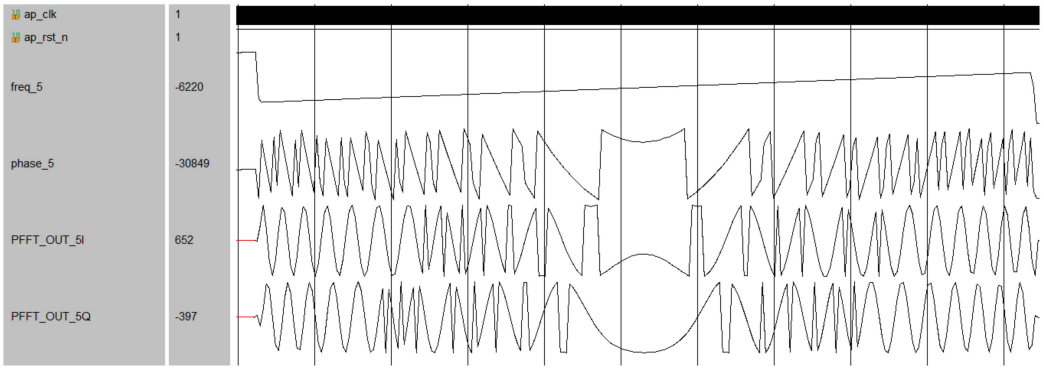


Figure 19. The results of instantaneous frequency extraction.

5. Conclusions

In this paper, an improved channel decision method has been proposed to address the issue of false signals in communication receivers based on the theory of blind-spot-free reception in a polyphase filtering structure. The method combines instantaneous feature extraction, the adaptive detection of signal amplitude, and phase difference frequency measurement algorithms to transform the process of eliminating false signals into extracting amplitude, phase, and frequency information from the output signal. This new approach improves the discrimination of useful signals and is more easily implemented on FPGA. Additionally, the hardware implementation of the receiver is optimized for speed and latency using HLS high-level synthesis technology, which significantly shortens the development cycle compared to traditional FPGA development processes. The simulation results confirm the practical value of this method for wideband digital receivers.

Author Contributions: Conceptualization, Z.J.; methodology, Z.J.; formal analysis, Z.J.; investigation, Z.J.; writing—original draft preparation, Z.J.; writing—review and editing, Z.J.; supervision, H.L.; project administration, H.L.; funding acquisition, H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China (Grant No. U2241221).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Man, J.; Li, W.; Wang, H.; Ma, W. On the technology of frequency hopping communication network-station selection. In Proceedings of the 2021 International Conference on Electronics, Circuits and Information Engineering (ECIE), Zhengzhou, China, 22–24 January 2021; pp. 35–41.
2. Chen, G.; Liu, L. Overview of Carrier frequency concealment Radar Signal Design. In Proceedings of the 2018 International Conference on Mechanical, Electronic, Control and Automation Engineering (MECAE 2018), Qingdao, China, 30–31 March 2018; Atlantis Press: Amsterdam, The Netherlands, 2018; pp. 497–500.
3. Obadi, A.B.; Soh, P.J.; Aldayel, O.; Al-Doori, M.H.; Mercuri, M.; Schreurs, D. A survey on vital signs detection using radar techniques and processing with FPGA implementation. *IEEE Circuits Syst. Mag.* **2021**, *21*, 41–74. [CrossRef]
4. Ghelfi, P.; Scotti, F.; Onori, D.; Bogoni, A. Photonics for ultrawideband RF spectral analysis in electronic warfare applications. *IEEE J. Sel. Top. Quantum Electron.* **2019**, *25*, 1–9. [CrossRef]
5. Sharma, P.; Sarma, K.K.; Mastorakis, N.E. Artificial intelligence aided electronic warfare systems-recent trends and evolving applications. *IEEE Access* **2020**, *8*, 224761–224780. [CrossRef]
6. Wang, H.; Lü, Y. A digital instantaneous frequency measurement receiver based on sub-Nyquist sampling. In Proceedings of the IEEE 10th International Conference On Signal Processing Proceedings, Beijing, China, 24–28 October 2010; pp. 1902–1905.
7. Udayakumar, R.; Khanaa, V.; Saravanan, T. Chromatic dispersion compensation in optical fiber communication system and its simulation. *Indian J. Sci. Technol.* **2013**, *6*, 4762–4766. [CrossRef]
8. Misra, V.; Bozkurt, A.; Calhoun, B.; Jackson, T.; Jur, J.S.; Lach, J.; Lee, B.; Muth, J.; Oralkan, Ö.; Öztürk, M.; et al. Flexible technologies for self-powered wearable health and environmental sensing. *Proc. IEEE* **2015**, *103*, 665–681. [CrossRef]
9. Zhang, Z.; Li, H.; Zhang, S.; Liu, Y. Analog-to-digital converters using photonic technology. *Chin. Sci. Bull.* **2014**, *59*, 2666–2671. [CrossRef]
10. Cordeiro, R.F.; Prata, A.; Oliveira, A.S.R.; Vieira, J.M.N.; De Carvalho, N.B. Agile all-digital RF transceiver implemented in FPGA. *IEEE Trans. Microw. Theory Tech.* **2017**, *65*, 4229–4240. [CrossRef]
11. Ma, D.; Du, Y.; Xu, C.Z.; Shao, Y.-Z.; Wang, S.-J.; Ping, Y.-F. Research on Total Probability Digital Channelization Technique without Blind Zone. In *Wireless and Satellite Systems, Proceedings of the11th EAI International Conference, WiSATS 2020, Nanjing, China, 17–18 September 2020*; Proceedings, Part II; Springer International Publishing: Cham, Switzerland, 2021; pp. 480–486.
12. Comoretto, G.; Chiello, R.; Roberts, M.; Halsall, R.; Adami, K.Z.; Alderighi, M.; Aminaei, A.; Baker, J.; Belli, C.; Chiarucci, S.; et al. The signal processing firmware for the low frequency aperture array. *J. Astron. Instrum.* **2017**, *6*, 1641015. [CrossRef]
13. Lu, W.-X.; Xu, P.; Cao, Y.-J. Modified Design of Digital Channelized Structure and Detection Method. *Mod. Def. Technol.* **2018**, *46*, 73–79.
14. Abo-Zahhad, M. ECG signal compression using discrete wavelet transform. In *Discrete Wavelet Transforms: Theory and Applications*; IntechOpen: London, UK, 2011; pp. 143–168.
15. Zheng, J.; Wu, W.; Zhao, Y.; Yuan, S.; Dong, R.; Zhang, L.; Fu, H. A Parallel Approach for Oil Palm Tree Detection on a SW26010 Many-Core Processor. In Proceedings of the IGARSS 2022–2022 IEEE International Geoscience and Remote Sensing Symposium, Kuala Lumpur, Malaysia, 17–22 July 2022.
16. Li, W.; He, C.; Fu, H.; Zheng, J.; Dong, R.; Xia, M.; Yu, L.; Luk, W. A Real-Time Tree Crown Detection Approach for Large-Scale Remote Sensing Images on FPGAs. *Remote Sens.* **2019**, *11*, 1025. [CrossRef]
17. Tian, Q.; Jiang, A.; Bi, B. Design and FPGA Implementation of a Digital Channelized Receiver. In Proceedings of the 2022 IEEE 2nd International Conference on Information Communication and Software Engineering (ICICSE), Chongqing, China, 18–20 March 2022; pp. 200–204. [CrossRef]
18. Wang, Y.; Zhang, M.; Ji, S.; Wang, J. Design of Multi-channel Real-time Signal Acquisition System. *J. Phys. Conf. Ser.* **2022**, *2189*, 012007. [CrossRef]
19. Özkan, M.A.; Pérard-Gayot, A.; Membarth, R.; Slusallek, P.; Leißa, R.; Hack, S.; Teich, J.; Hannig, F. AnyHLS: High-level synthesis with partial evaluation. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2020**, *39*, 3202–3214. [CrossRef]
20. Wang, H.; Lü, Y.Y. A digital channelized receiver architecture with low calculation cost. In Proceedings of the 2010 Second Pacific-Asia Conference on Circuits, Communications and System, Beijing, China, 1–2 August 2010; Volume 1, pp. 1–4.
21. Bai, F.; Gagar, D.; Foote, P.; Zhao, Y. Comparison of alternatives to amplitude thresholding for onset detection of acoustic emission signals. *Mech. Syst. Signal Process.* **2017**, *84*, 717–730. [CrossRef]
22. Mohamed, S.M.; Sayed, W.S.; Radwan, A.G.; Said, L.A. FPGA implementation of reconfigurable CORDIC algorithm and a memristive chaotic system with transcendental nonlinearities. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2022**, *69*, 2885–2892. [CrossRef]

23. Li, T.; Chen, Z. A wideband reconnaissance receiver design based on real-time spectrum analysis technology. In Proceedings of the 2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, China, 25–26 March 2017; pp. 1463–1467.
24. Cong, J.; Liu, B.; Neuendorffer, S.; Noguera, J.; Vissers, K.; Zhang, Z. High-level synthesis for FPGAs: From prototyping to deployment. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2011**, *30*, 473–491. [CrossRef]
25. Liu, X.; Wang, Z.K.; Deng, Q.X. Design and FPGA implementation of a reconfigurable 1024-channel channelization architecture for SDR application. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2016**, *24*, 2449–2461. [CrossRef]
26. Rauf, A.; Pasha, M.A.; Masud, S. Towards design and automation of a scalable split-radix FFT processor for high throughput applications. *Microprocess. Microsyst.* **2019**, *65*, 148–157. [CrossRef]
27. Roslidar, R.; Zulfikar, Z. MATLAB based Design for an 8-point Discrete Fourier Transform formed on Products of Rademacher Functions. *J. Telecommun. Electron. Comput. Eng.* **2018**, *10*, 39–43.
28. Rachana, M.K.; Manju, V.M. Study on Area Efficient Radix-2 FFT Architecture to Process Twin Data Streams for High Speed Real Time Application. *Indian J. Emerg. Electron. Comput. Commun.* **2017**, *4*, 638–643.
29. Murali, A.; Kakarla, H.K.; Anitha Priyadarshini, G.M. Improved design debugging architecture using low power serial communication protocols for signal processing applications. *Int. J. Speech Technol.* **2021**, *24*, 291–302. [CrossRef]