Journal of Intelligent Systems and Control
https://www.acadlore.com/journals/JISC

Design and Implementation of FPGA for Digital Channelization Processing

Ming Yin[1], Yanlei Fu[2*], Jiefeng Mei[1]

[1] School of Information Science and Technology, Qingdao University of Science and Technology, 266100 Qingdao, China
[2] Gaomi Campus, Qingdao University of Science and Technology, 261500 Weifang, China

[*] Correspondence: Yanlei Fu (gm127@qust.edu.com)

Received: 07-02-2024 **Revised:** 09-15-2024 **Accepted:** 09-23-2024

Citation: M. Yin, Y. L. Fu, and J. F. Mei, "Design and implementation of FPGA for digital channelization processing," *J. Intell Syst. Control*, vol. 3, no. 3, pp. 160–173, 2024. https://doi.org/10.56578/jisc030303.

Abstract: To address the rate mismatch between high-bandwidth, high-sampling-rate analog-to-digital converters (ADCs) and low-bandwidth, low-sampling-rate baseband processors, digital signal processing techniques were employed to enable the parallel processing of broadband signals. The broadband signals were decomposed into multiple narrowband channels, facilitating parallel processing and frequency-selective analysis of signals. The validity of the principle was verified through MATLAB modeling. A digital channelization Register Transfer Level (RTL) model was constructed on a Field-Programmable Gate Array (FPGA) using Verilog Hardware Description Language (HDL), implementing a pipelined parallel processing mechanism. The computational efficiency in Fast Fourier Transform (FFT) operations was improved by optimizing the processing flow. A digital channelization receiver application test board was developed using a domestically produced FMQL45T900 FPGA as the core component. Practical applications confirmed the correctness of the approach, with significant improvements in power efficiency compared to methods reported in existing literature, thereby enhancing overall parallel processing performance. This method demonstrates broad applicability in fields such as military communications, broadcasting, radar navigation systems, and more.

Keywords: FPGA; Digital channelization; FIR filter

1 Introduction

Electronic warfare receivers are required to possess a wide instantaneous bandwidth, exceptional demodulation sensitivity and frequency resolution, extensive dynamic range, and the ability to process multiple signals in parallel. Against this backdrop, channelized receivers have emerged as an ideal solution due to their unique performance advantages, making them well-suited to meet the demands of modern battlefield environments. Digital channelization is a method that employs encoding and decoding techniques to process digital communication signals, thereby enhancing transmission quality and reliability [1]. In recent years, with the continuous advancement of communication technologies and the expansion of application scenarios, the importance of digital channelization and digital filters in communication systems, the Internet of Things, satellite communications, and intelligent transportation has significantly increased [2].

FPGA, as a flexible and programmable hardware platform, provides an efficient means for implementing digital channelization algorithms. FPGA-based digital channelization techniques represent an effective signal processing approach capable of decomposing a broadband signal with substantial instantaneous bandwidth into multiple narrowband signals, which are then allocated to different channels. The number of channels is typically predetermined, such as 32-channel processing mentioned in this study.

Thamizharasan and Kasthuri [3] proposed an efficient FPGA-based digital down-conversion design tailored to software radio applications. Zhang et al. [4] introduced a design approach that incorporates undersampling and frequency estimation in channelization applications. Datta and Dutta [5] designed the efficient implementation of a half-band filter on the FPGA. Kim et al. [6] used digital down-conversion for equivalent time-domain signal sampling in complex periodic signals. Furthermore, Zhang et al. [7] presented an ultra-wideband digital back-end system for the QTT, employing a Polyphase Filter Bank (PFB) algorithm. Zhang et al. [8] investigated channelization techniques for radio astronomical wideband signals using oversampled PFBs. Liu et al. [9] proposed a frequency measurement approach for microwave signals over a wide frequency range, leveraging an optical frequency comb combined with a channelization method. Huiskamp et al. [10] introduced a method for reconstructing aliased frequency spectra by employing multiple sample rates. Song et al. [11] presented an optimized compressed spectrum sensing technique based on a multicoset sampler for multiband signals, approaching the sub-Nyquist boundary. Furthermore, Huang et al. [12] demonstrated parameter measurement for multiple exponentially damped sinusoids using sub-Nyquist sampling.

While these studies focus on the application of algorithms for digital processing, they lack validation and extension in practical engineering implementations. To address critical technological bottlenecks and reduce manufacturing costs, it is imperative to pursue independent development and domestically produced alternatives for core technologies. The FPGA design of the digital channelization receiver described in this study includes several key modules, namely the data preprocessing module, the Finite Impulse Response (FIR) filter bank module, the inverse FFT (IFFT) module, and the digital signal processing module. The data preprocessing module is responsible for introducing external signals into the system and performing integer-factor decimation on the input data, which reduces the sampling rate while transmitting the processed data to the FIR module. Subsequently, the FIR filter bank module processes the signals by filtering them according to their frequency characteristics and performing necessary spectrum shifting. Next, an IFFT module, implemented using butterfly operations, performs the 32-channel IFFT, converting signals from the frequency domain to the time domain. Finally, the digital signal processing module applies quantization and other processing steps to the output signals to meet specific output requirements. Through the integration and coordinated operation of these modules, the overall digital channelization workflow is achieved. This process effectively decomposes the original signal into multiple channels for parallel processing, ultimately accomplishing digital channelization.

2 Digital Channelization Principle

Digital channelization is achieved through the reception of bandpass signals using FIR filters. During the channelization process, digital processing techniques such as FFT were employed to enable efficient and accurate signal reception and processing. When handling data sampled by the ADC, the data acquisition rate of the ADC differed from the processing rate of the receiver, necessitating the consideration of data buffering and processing during transmission.

2.1 Multi-Rate Data Conversion

To reduce the data rate obtained from the ADC, integer-factor decimation is commonly applied. Integer decimation refers to the process of resampling the original sequence $x(n)$ by selecting every N-th sample point (where N is an integer), resulting in a new sequence $x_N(m)$ with a sampling rate $f_s' = f_s/N$ [13]. This can be expressed as follows:

$$x_N(m) = x(mN)$$

Assuming the sampling rate of the original sequence is f_s, with a sampling interval of T, the Fourier transform of the original sequence is given by:

$$\mathrm{X}\left(\mathrm{e}^{j\varpi}\right)\big|_{\varpi=\Omega T} = \hat{X}_a(j\Omega) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X_a\left(j\Omega - j\frac{2\pi}{T}k\right) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X_a\left(j\frac{\varpi-2\pi}{T}k\right)$$

The sampling rate of the decimated sequence $x_N(m)$ became $f'_s = f_s/N$, with a corresponding sampling interval of $T' = NT$. The Fourier transform of the sequence $x_N(m)$ can be expressed as follows:

$$\mathrm{X}\left(\mathrm{e}^{j\varpi'}\right)\Big|_{\varpi'=\Omega T'} = \hat{X}_a'(j\Omega) = \frac{1}{T'}\sum_{k=-\infty}^{\infty} X_a\left(j\Omega - j\frac{2\pi}{T'}k\right) = \frac{1}{NT}\sum_{k=-\infty}^{\infty} X_a\left(j\frac{\varpi'-2\pi}{T}k\right)$$

It can be observed from these expressions that integer-factor decimation improves the frequency-domain resolution of the ADC signal. This enhanced resolution is advantageous for subsequent digital processing workflows [14].

2.2 Digital Quadrature Downconversion

Digital quadrature downconversion is a widely used technique in communication systems, primarily employed to enhance spectrum utilization. In communication scenarios, signals transmitted through the air are typically real-valued [15]. Upon reception, such signals must be demodulated to convert them into baseband frequencies. During this process, the signal is decomposed into two orthogonal components: the in-phase component (I) and the quadrature component (Q), collectively referred to as IQ components. The purpose of this transformation is to facilitate efficient signal processing and enable subsequent processors or demodulators to interpret and handle the signal more effectively. Through IQ components, the receiver is capable of extracting critical information from the original signal, ensuring accurate data transmission and processing [16]. Consequently, digital quadrature downconversion plays a pivotal role in modern communication systems by enabling higher spectrum efficiency and reliable data transmission.

Digital quadrature downconversion typically employs $\cos(\omega_0 n)$ and $-\sin(\omega_0 n)$ to perform frequency translation. The original input signal is multiplied by these transformation factors to convert the radio frequency (RF) signal into I and Q signals. A structural diagram of this process is illustrated in Figure 1.

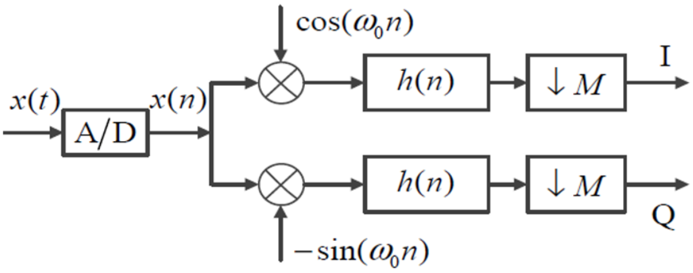


Figure 1. Structural diagram of digital quadrature downconversion

2.3 Channelization Structure Based on Polyphase Filtering

A digital channelization receiver based on a polyphase filter structure was designed, with the core principle being the use of a parallel sampling digital structure featuring different frequency mixing coefficients to achieve channelization in the frequency domain, thereby generating baseband signals [17]. Compared to directly processing high-speed ADC data through multiplication with mixing factors, this approach imposes lower requirements on mixers, making it more resource-efficient and easier to implement [18]. However, the increase in the number of channels, with each channel requiring a unique mixing factor, leads to higher hardware resource consumption [19].

To address this challenge, a channelization structure based on polyphase filtering was adopted for signal processing. This structure performs exceptionally well when handling multi-channel inputs and effectively reduces hardware resource consumption. The structural diagram of a polyphase filter [20] is shown in Figure 2. By utilizing polyphase filters, the receiver achieves more efficient processing of input signals, improving system performance and scalability.

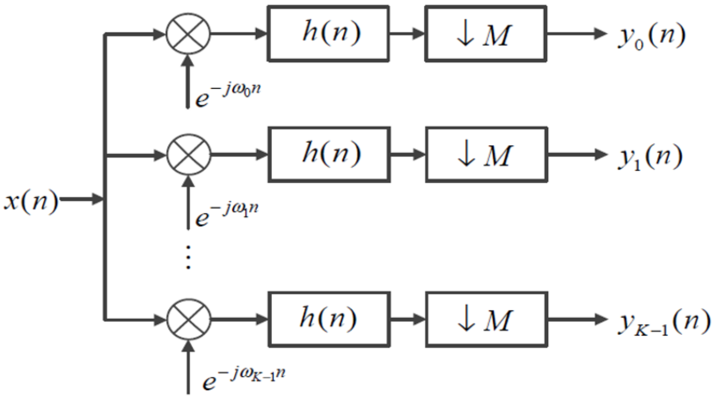


Figure 2. Structural diagram of channelization

3 Digital Channelization Receiver Design

3.1 Overall Framework

The FPGA module design for the digital channelization receiver comprises the following key components: the data preprocessing module, the filter bank module, the IFFT module, and the signal processing output module.

The data preprocessing module is responsible for converting serial input data into a parallel form, facilitating subsequent processing stages. The FIR filter bank module is then utilized to filter the input signals, eliminating noise and unwanted frequency components to enhance signal quality. The IFFT module performs the IFFT, converting frequency-domain signals back into the time domain for further processing. Finally, the signal processing output module applies necessary operations such as demodulation, decryption, or other signal processing algorithms [10].

Channel division can be categorized into uniform channelization and non-uniform channelization. In this design, a uniform channelization method was adopted for the filter bank design. The structural diagram of the system is shown in Figure 3.

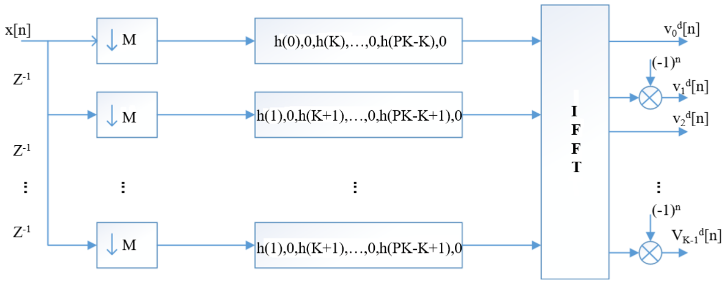

x[n]
Z-1
Z-1
Z-1
↓M
↓M
↓M
h(0),0,h(K),…,0,h(PK-K),0
h(1),0,h(K+1),…,0,h(PK-K+1),0
h(1),0,h(K+1),…,0,h(PK-K+1),0
IFFT
v0d[n]
(-1)n
v1d[n]
v2d[n]
(-1)n
VK-1d[n]

Figure 3. Channelization structure

Integer-factor decimation was achieved using serial-to-parallel conversion techniques to preprocess the acquired data. This technique allows high-speed serial data streams to be decomposed into multiple low-speed parallel data streams, thereby enabling data downsampling without any loss of information. This approach is particularly significant in digital signal processing as it facilitates the efficient management and handling of large data streams. The overall RTL framework for the FPGA design of digital channelization is illustrated in Figure 4.

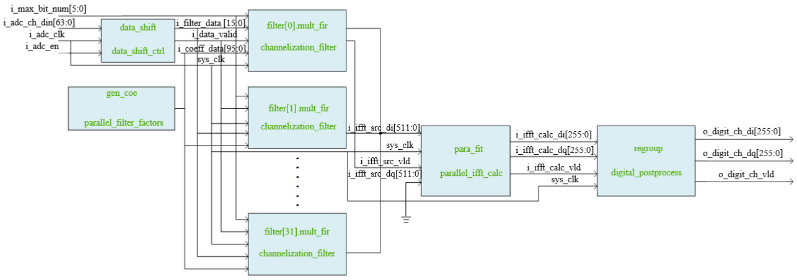


Figure 4. RTL diagram of the FPGA design for channelization

3.2 Data Preprocessing Module

To ensure the effective application of FIR filtering in subsequent stages, appropriate data preprocessing must be performed. The core objective of the preprocessing stage is to implement serial-to-parallel conversion, facilitating data rate reduction.

Integer-factor decimation was employed to achieve the downsampling of data acquired by the ADC. In FPGA implementations, this process is typically realized using serial-to-parallel conversion techniques [20]. The primary task of this module is to shift the ADC data and convert it into serial data with a specific bit width. During this process, due to the integer-factor relationship inherent in decimation, overlapping of output serial data occurs. For instance, if the output serial data has a bit width of 512 bits, there will be an overlap of 256 bits between consecutive segments. The workflow of the data preprocessing module is illustrated in Figure 5.

The core of implementing this process lies in the precise control of the flag bit, enabling the orderly shifting of input ADC data into registers to ultimately construct a complete 512-bit serial data stream. This sophisticated design not only optimizes the handling of high-speed data inputs but also fundamentally ensures the accuracy, integrity, and completeness of the data. Through this approach, the FPGA is able to effectively manage high-speed data streams from RF signals, providing a robust data foundation for subsequent signal processing and demodulation.

The RTL diagram of the data preprocessing module, as generated in Vivado, is shown in Figure 6.

3.3 FIR Filter Bank Design

From the perspective of hardware implementation, achieving an ideal sharp cutoff characteristic in filter design is challenging. In practical applications, a very narrow transition band significantly increases filter complexity, leading to a higher number of required multipliers and potentially extending filtering processing time. These factors considerably increase hardware resource consumption and reduce data processing rates [21].

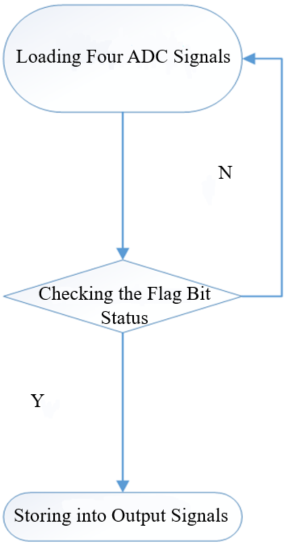


Figure 5. Workflow of serial-to-parallel data conversion

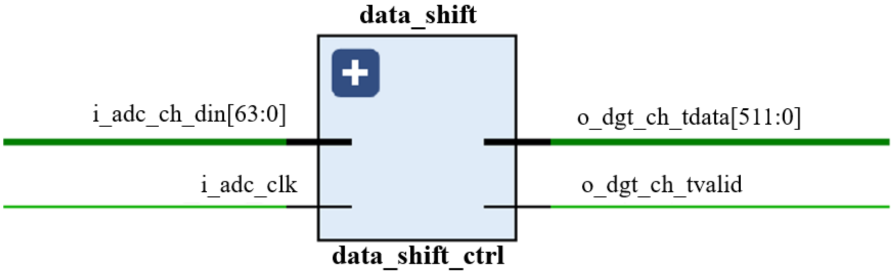


Figure 6. RTL diagram of the data preprocessing module

To ensure that no critical information is missed, a filter bank with 50% overlap was employed for filtering operations. This design enhances the comprehensiveness of information capture, ensuring continuity and accuracy during signal monitoring and minimizing the risk of data loss or omissions. Additionally, the transition band of each filter was designed to be relatively wide. This approach effectively reduces the order of the FIR filters, simplifying the filter structure and significantly lowering computational complexity and resource utilization, thereby efficiently utilizing hardware resources.

Table 1. Channel allocation and filter numbers

Filter Number	Start Frequency (MHz)	Centre Frequency (MHz)	Cutoff Frequency (MHz)
0	520	512	504
1	504	496	488
2	488	480	472
3	472	464	456
4	456	448	440
5	440	432	424
6	424	416	408
7	408	400	392
8	392	384	376
9	376	368	360
10	360	352	344
11	344	336	328
12	328	320	312
13	312	304	296
14	296	288	280
15	280	272	264

By employing this filter bank design, hardware resources were utilized efficiently while maintaining system performance. Additionally, the approach ensures full coverage of all frequencies within the monitoring bandwidth. This method is widely adopted in digital signal processing, particularly in scenarios where efficient signal processing is required under limited hardware resources. Table 1 shows the channel allocation and filter numbers.

The bandpass filter bank was implemented by modulating a prototype low-pass filter with 192 taps. Based on the selected design parameter K=32, the number of phases P in each PFB can be calculated, with P=N/K. Thus, 192/32=6. Each phase filter coefficient was extended by inserting zeros, specifically two zeros between every coefficient, resulting in 12 coefficients for each phase filter.

In this PFB design, each coefficient of the prototype filter was reused in a specific pattern, with zero values inserted between repetitions of the coefficients. This approach not only conserves hardware resources but also enhances the flexibility and response speed of the filter. By processing input signals through these polyphase filters, uniform distribution of the data stream across all phases is ensured, enabling efficient frequency bandwidth utilization and signal reconstruction.

The data processing workflow demonstrates how input data is managed through these polyphase filters and how the processed data is combined to form the final output signal. In practical applications, this polyphase filter structure is commonly employed for efficient signal demodulation and bandwidth management, particularly in frequency-selective signal processing and communication systems. The implementation details and optimization strategies of each FIR filter significantly influence the overall performance and efficiency of the system. Figure 7 shows the filter structure.

When the FIR_STEP parameter was set to 12 and the number of filters was 32, each FIR filter contained 192 coefficients, with each coefficient occupying 16 bits. These coefficients were generated using the filterDesigner tool in MATLAB. The coefficients were divided into 32 groups, each containing 12 coefficients, with zero-padding applied between coefficients. The RTL implementation diagram of the FIR filter bank module is shown in Figure 8.

The required tap coefficients for each filter were not described individually. Instead, the FIR filters were designed in MATLAB, and their functions were generated. These tap functions were directly written into Verilog HDL files, serving as the tap sources for the FIR filter bank.

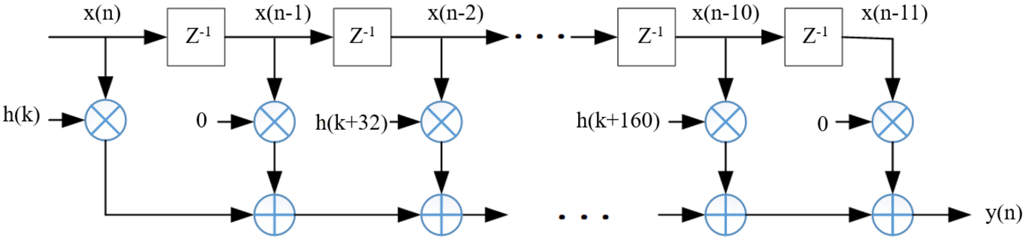


Figure 7. Filter structure

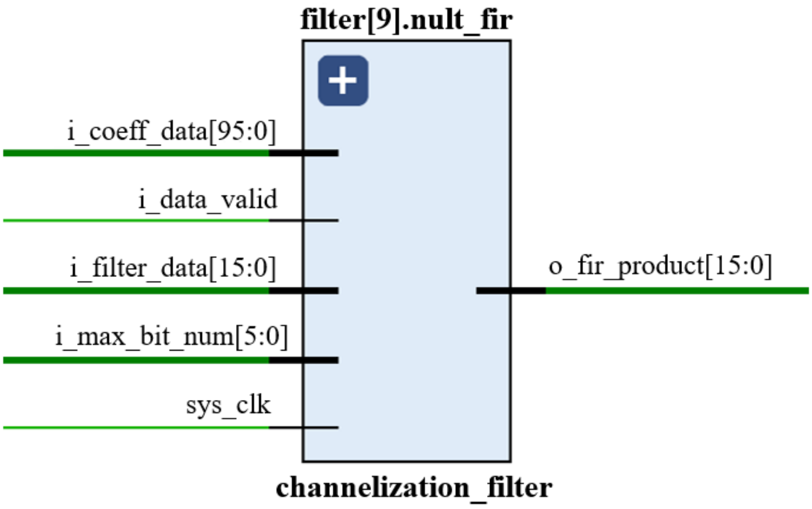


Figure 8. RTL implementation of the FIR filter bank

3.4 IFFT Module

The IFFT is a critical technique used to convert frequency-domain signals back into the time domain. In the implementation of the IFFT, a practical approach involves first removing the complex conjugate of the signal, then performing a FFT, followed by taking the complex conjugate of the result and multiplying it by a coefficient $\left(\frac{1}{N}\right)$.

For processing ADC sampling data, PFBs were employed for filtering. This approach enables the simultaneous output of 32 filtered signals, achieving efficient parallel processing. The utilization of this filtering technique not only enhances the speed of data processing but also ensures the accuracy and quality of signal handling. To address the issue of excessive data throughput caused by serial input, a fully parallel IFFT operation based on radix-2 time-decimation was adopted. This design ensures efficient handling of high-speed data streams while maintaining the fidelity of the transformed signals, making it particularly suitable for high-performance digital signal processing applications. In this design, the IFFT module features 32 inputs and is tasked with executing 32-point parallel IFFT operations. By leveraging the properties of the FFT, a 32-point Inverse Discrete Fourier Transform (IDFT) can be decomposed into a series of smaller-scale IDFT computations. Specifically, a 32-point IDFT can be divided into two 16-point IDFTs. This decomposition process can be recursively continued until the original 32-point IDFT is reduced to 16 computations, each comprising 2-point IDFTs. This decomposition methodology significantly simplifies computational complexity and enhances operational efficiency. Based on this principle, Decimation-In-Time for the IFFT (DIT-IFFT) can be applied to the 32-channel data output from the filter bank. This approach enables efficient handling and analysis of the data, further optimizing the overall signal processing workflow. The RTL diagram of the 32-point parallel IFFT module is shown in Figure 9.

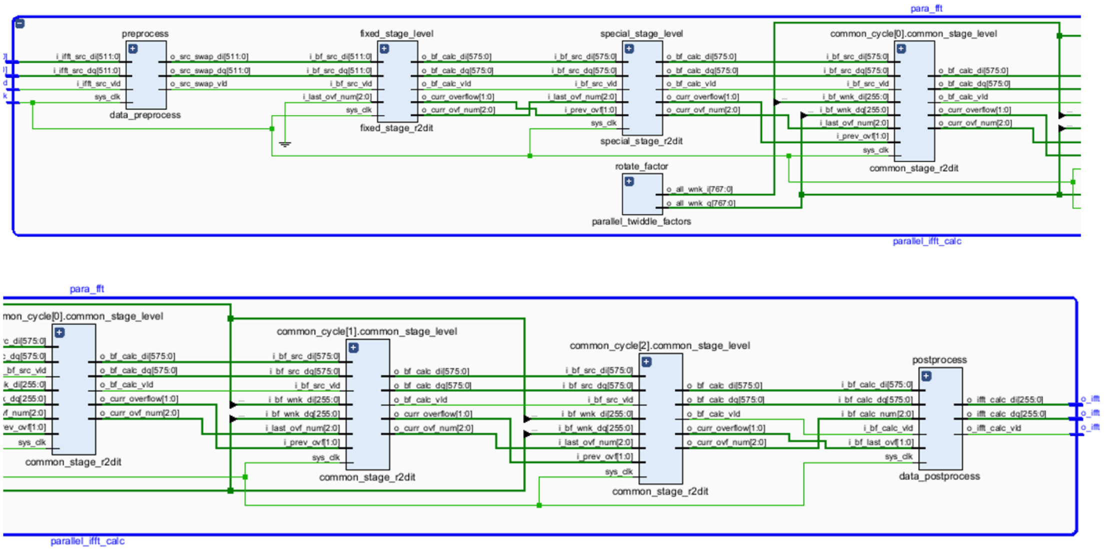


Figure 9. RTL diagram of the 32-point parallel IFFT module

3.5 Digital Signal Processing Module

This module performs specific operations on the output data from the IFFT, focusing on the processing of odd-numbered channels. For the output data of the n-th odd channel, a multiplication operation by the n-th power of -1 is applied, where n represents the output data of the n-th channel.

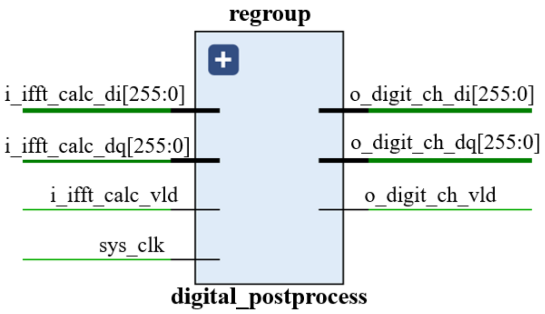


Figure 10. RTL design of the digital signal processing module

In digital quadrature demodulation, the exponential term in the product $e^{-jw_k Mn}$ can be expressed as $(\cos k\pi - j \times \sin k\pi)^n$, where $F = \frac{K}{M} = 2; k$ denotes the k-th phase result from the filter; and n represents the output data of the n-th channel. This indicates that the data from the n-th channel undergoes a multiplication operation corresponding to $(\text{-}1)^{\text{n}}$. Specifically, the data from odd-numbered channels are processed by × (-1), while the data from even-numbered channels remain unchanged. Following the operation, all signals are output together. In practical terms, the data from channels 0, 2, ..., 14 are left unaltered, while the data from channels 1, 3, ..., 15 are multiplied by (-1) before being output. The purpose of this operation is to adjust the channels to ensure correct processing or demodulation in subsequent stages. The RTL design of the digital signal processing module is shown in Figure 10.

4 Digital Channelization Simulation

4.1 MATLAB Design of the Digital Channelization Receiver

The key tasks in the MATLAB simulation include generating the tap coefficients for the FIR filter bank and producing the signals required for the digital channelization receiver.

In this module, the FIR filter tap coefficients were generated based on the design requirements. These coefficients define the frequency response characteristics of the filters used for processing the input signals, including parameters such as filter type and cutoff frequency. The generation of these coefficients must consider the filter design specifications and performance criteria to meet the system requirements.

Additionally, MATLAB was employed to simulate the generation of analog-to-digital (AD) signals required for digital channelization. The ADC acquisition process was mimicked, and the data was exported into a text file. These data served as the ADC acquisition inputs during the digital channelization simulation.

To achieve efficient data processing, a uniform channelization strategy was selected, with the number of subchannels set to K = 32. To capture and analyze detailed information within the signal, an ADC sampling frequency of 512 MHz was adopted. Based on this sampling frequency and the number of subchannels, the bandwidth of each subchannel was precisely calculated to be 16 MHz, while the total processing bandwidth extended up to 32 MHz, ensuring broad signal coverage and high-resolution capability. To optimize signal quality, particularly during the filtering stage, a 192-tap low-pass filter was specifically designed. This filter plays a critical role in the digital channelization system, effectively eliminating high-frequency noise and interference to ensure accuracy and reliability during subsequent signal processing. This well-designed filtering scheme provides a robust foundation for the entire signal processing workflow. The Filter Designer tool in MATLAB was used to design the filter. The filter function was then exported and incorporated into the FIR module, where it served as the basis for FIR filter design.

Instead of using ADC for data acquisition, MATLAB was employed to simulate the acquisition process. In this simulation, ADC data was generated in MATLAB and an ADC sampling frequency of 512 MHz was set, with the carrier frequency of the signal to be captured set at 320 MHz and the simulated ADC sampling bandwidth set to 20 MHz.

Using MATLAB, a function was employed to generate a modulated signal centred around a carrier frequency of 320 MHz with a bandwidth of 20 MHz. This signal was then sampled at a frequency of 512 MHz, indicating that the signal was sampled 512 million times per second. Following the sampling process, the resulting digital signal was converted into binary data and saved to a text file for subsequent simulation. Figure 11 shows the amplitude-frequency spectrum.

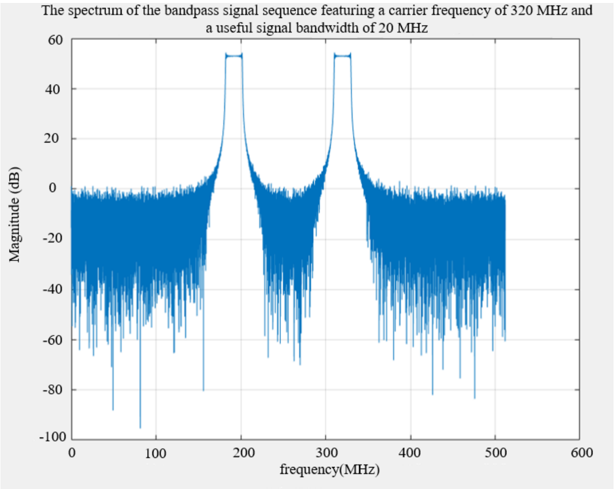


Figure 11. Amplitude-frequency spectrum

4.2 Digital Channelization Receiver Simulation

4.2.1 Simulation of the data preprocessing module

The primary task of the data preprocessing module is to convert parallel AD data into a 512-bit serial data stream. This module plays a critical role in the signal processing system by transforming high-speed parallel data into a serial format that is easier to handle and transmit, thus facilitating subsequent signal processing stages.

In this module, parallel AD data, typically consisting of a series of samples acquired from the analog signal, was first received. The ADC data used in this simulation was generated by MATLAB, representing a radio-frequency signal. The data was directly read from the previously generated text file. Subsequently, the parallel data was converted into a 512-bit serial data stream. The simulation results of this module are presented in Figure 12.

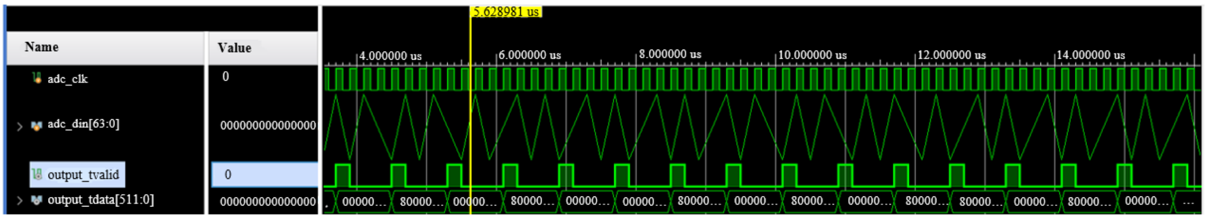


Figure 12. Simulation results of the data preprocessing module

The adc_clk signal serves as the clock signal used to control AD data acquisition. The $output\_tvalid$ flag is a status indicator that signifies the completion of the serial-to-parallel data conversion. The adc_din signal represents the input data from the ADC , while $output\_tdata$ refers to the converted data output after the serial-to-parallel transformation. In summary, when the adc_clk is triggered, AD data is input through adc_din. After undergoing serial-to-parallel conversion, the data is output through $output\_tdata$ once the $output\_tvalid$ flag becomes active.

4.2.2 Simulation of the FIR filter bank

The primary function of the FIR filter bank module is to perform filtering and mixing operations on the input signal, removing components outside the predefined frequency range to ensure that the output signal meets specific requirements. Additionally, this module shifts the spectrum of the input signal to the baseband while generating the corresponding I and Q components. These operations provide essential preprocessing for the subsequent IFFT stage. Through precise filtering by the FIR filter bank, the processed signal can become cleaner and more aligned with the system requirements, thereby improving overall system performance. The simulation results of this module are shown in Figure 13.

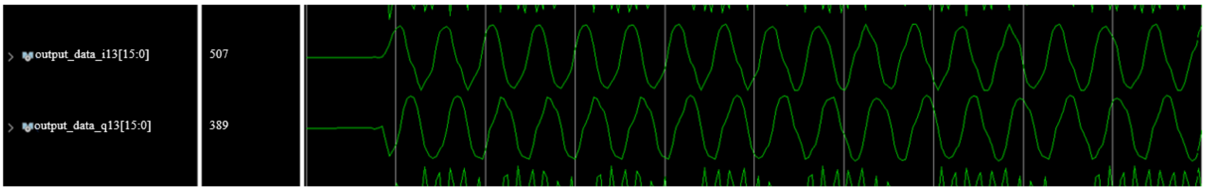


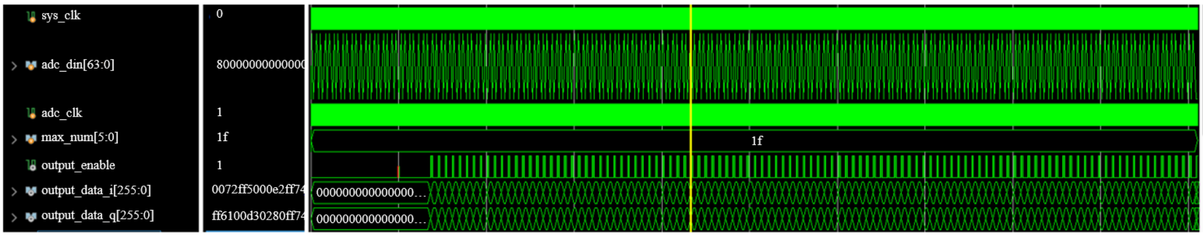
Figure 13. Simulation of a single filter

$output\_data\_i$ [13] and $output\_data\_q$ [13] represent the I and Q components of the output from the 13th channel, respectively. The figure illustrates the output results of the 13th input signal after processing by the FIR filter bank. It can be observed that, after filtering, the output signal retains only the required 320 MHz component, while unwanted frequency components are effectively removed. This demonstrates that the FIR filter bank successfully filters out signals outside the specified range and shifts the spectrum of the desired signal to the baseband.

4.2.3 Simulation of the digital channelization receiver

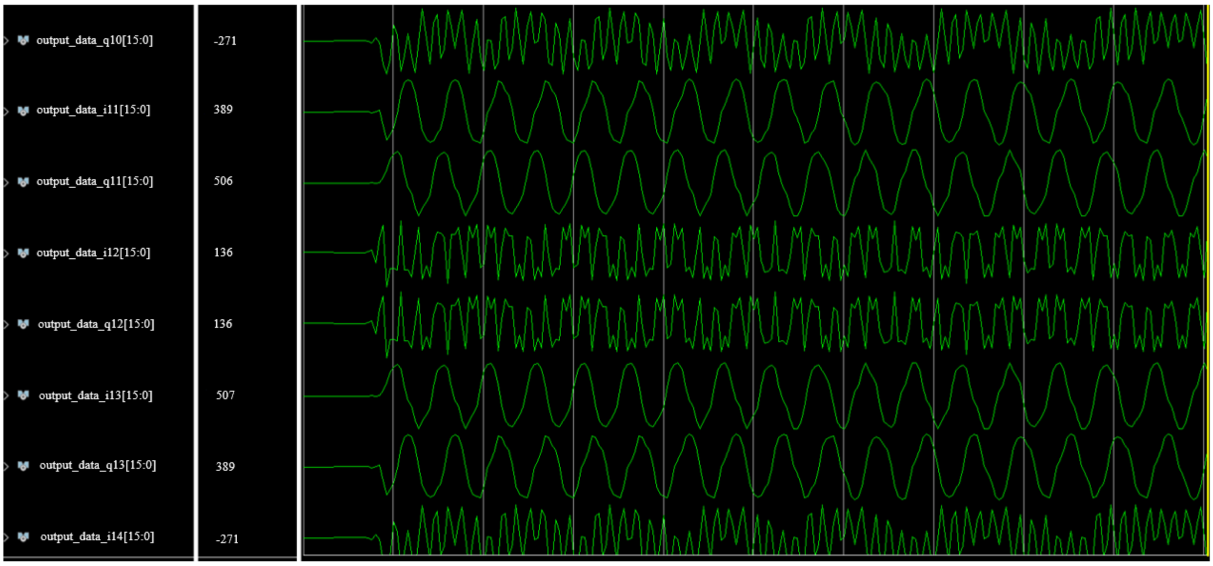
The primary function of the digital channelization receiver is to convert input signals with a large instantaneous bandwidth into output signals with smaller instantaneous bandwidths. This conversion process not only conserves communication resources and improves signal transmission efficiency but also ensures that the signals can be accurately reconstructed and decoded at the receiver end, enabling reliable communication and data transmission.

In this simulation, the input excitation signal was generated using MATLAB and saved as a TXT file to serve as the ADC-acquired data. The overall simulation results of the digital channelization receiver are shown in Figure 14. Figure 15 shows partial channel output results.



sys_clk 0
adc_din[63:0] 8000000000000(
adc_clk 1
max_num[5:0] 1f
output_enable 1
output_data_i[255:0] 0072ff5000e2ff74
output_data_q[255:0] ff6100d30280ff74

Figure 14. Overall simulation results of the digital channelization receiver



output_data_q10[15:0] -271
output_data_i11[15:0] 389
output_data_q11[15:0] 506
output_data_i12[15:0] 136
output_data_q12[15:0] 136
output_data_i13[15:0] 507
output_data_q13[15:0] 389
output_data_i14[15:0] -271

Figure 15. Partial channel output results

The signals processed by the system included the input ADC signal, marked as adc_in. After processing, the system output 32 channels of serial data. The I components, representing the real part, were stored in $output\_data\_i$, while the Q components, representing the imaginary part, were stored in $output\_data\_q$. For different digital channels, specific output signals were identified as follows:

a) For the 11th channel, the I and Q components are represented by $output\_data\_i11$ and $output\_data\_q11$, respectively, corresponding to a channel bandwidth of 328 MHz to 344 MHz.

b) For the 12th channel, the I and Q components are represented by $output\_data\_i12$ and $output\_data\_q12$, respectively, corresponding to a channel bandwidth of 312 MHz to 328 MHz.

c) For the 13th channel, the I and Q components are represented by $output\_data\_i13$ and $output\_data\_q13$, respectively, corresponding to a channel bandwidth of 296 MHz to 312 MHz.

d) For the 14th channel, the I component is represented by $output\_data\_i14$, corresponding to a channel bandwidth of 280 MHz to 296 MHz.

e) For the 10th channel, the Q component is represented by $output\_data\_q10$, corresponding to a channel bandwidth of 344 MHz to 360 MHz.

Upon entering the digital channelization system, the input RF signal underwent a series of processing stages, including serial-to-parallel conversion, FIR filter bank processing, IFFT computation, and subsequent transformations. This sequence of operations transformed the original wideband RF signal into signals with narrower instantaneous bandwidths, enabling the receiving end to capture and analyze the transmitted signals more effectively. As shown in the results above, and based on the previously defined channel division, it is evident that the responses of the 11th and 13th channels were significantly stronger than those of other channels when the system processed signals acquired by the ADC. These simulation results confirm that the digital channelization receiver successfully completed the digital channelization process. The instantaneous bandwidth and sampling rate of the input signal were significantly reduced.

5 Test Results

The digital channelization processing was tested on the FMQL45T900 FPGA evaluation board developed by Fudan Microelectronics. A setup with five ADC channels was implemented, capturing five intermediate-frequency signals. A single RF signal was generated using a signal generator, split into five parallel signals via a power splitter, and input into the five ADC channels. On the FPGA board, the processes of AD sampling, data preprocessing, FIR filtering, and IFFT transformation were completed to extract corresponding frequency information. These operations provide the foundation for subsequent Fourier transformations, frequency acquisition, phase extraction, and other processing tasks. The domestic FMQL45 series FPGA by Fudan Microelectronics is fully compatible with the ZYNQ 7045 series developed by Xilinx.

The FMQL45T900 integrates two ARM Cortex-A9 cores and a Kintex 7 FPGA. The FPGA's programmable logic (PL) extends interfaces such as FMC, Ethernet, and I/O, while the processing system (PS) based on ARM supports extensions for networking, USB, and RS232 interfaces. The board is suitable for applications in image processing, vibration analysis, communication systems, radar front-end signal processing, and handheld device development. Figure 16 shows the test board. Figure 17 shows the workflow diagram for the digital channelization application.

- Number of channels: 5 parallel channels.
- ADC resolution: 16-bit resolution for low-speed channels.
- Maximum sampling rate: 250 Msps for low-speed channels.
- Acquisition clock: Supports both internal and external clocks.
- External trigger: Equipped with an external trigger interface.
- Synchronization precision: Better than 0.03 ns.
- Spurious-Free Dynamic Range (SFDR): Better than 85 dBFS for low-speed channels.
- Signal-to-Noise Ratio (SNR): Better than 70 dBFS for low-speed channels.

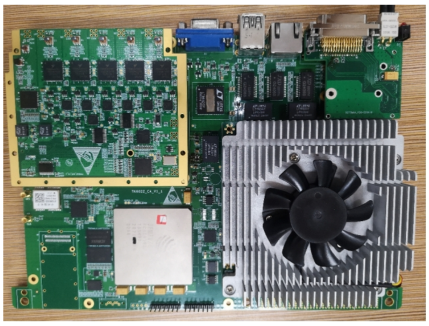


Figure 16. Test board

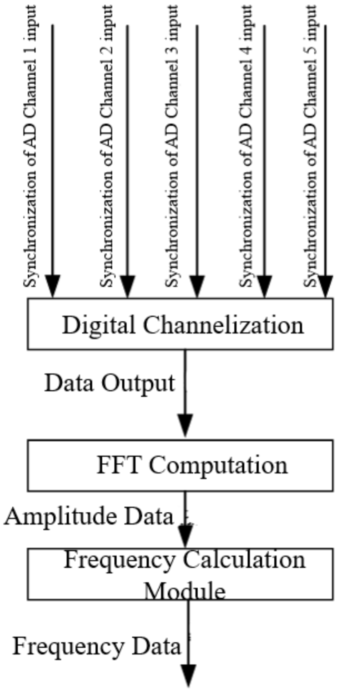

Synchronization of AD Channel 1 input
Synchronization of AD Channel 2 input
Synchronization of AD Channel 3 input
Synchronization of AD Channel 4 input
Synchronization of AD Channel 5 input
Digital Channelization
Data Output
FFT Computation
Amplitude Data
Frequency Calculation Module
Frequency Data

Figure 17. Workflow diagram for digital channelization application

The validation of the proposed system was performed using a test setup built around the domestic FMQL45T900 FPGA from Fudan Microelectronics. Intermediate-frequency signals were sampled, and wideband signals were processed in parallel across multiple channels. Amplitude data were obtained using FFT, and frequency information was derived using a frequency estimation algorithm. These tests verified the accuracy and effectiveness of the digital channelization receiver.

The proposed design was implemented on the FMQL45T900 FPGA, and the logic resource utilization is summarized in Table 2.

Table 2. Channel allocation and filter numbers

Parameter	Digital Channelization	DDC	DSP Devices
Slice registers	228	420	50
Slices	97	108	100
Slice LUT	132	135	50
LUT flip-flop pairs	280	410	62
Bonded IOB	65	40	107

The proposed method was compared with the approach presented by Zhang et al. [8] regarding area, frequency, and power consumption, as shown in Table 3.

Table 3. Performance comparison of architectures

Important Parameter	Zhang et al. [8]	Proposed Method
Slice registers (area)	1882	1266
Frequency (MHz)	472.4	550
Power	1022	985

It is evident from Table 2 that the proposed method reduces the required slice registers compared to the work by Zhang et al. [8]. Table 3 further illustrates that both power consumption and performance have been improved. Moreover, the proposed method demonstrates effective implementation on the domestic FMQL45T900 FPGA, providing a robust foundation for digital channelization processing in indigenous electronic, communication, and radar systems. This ensures independence from foreign chips, thereby mitigating risks associated with potential restrictions and delays in equipment development.

6 Conclusion

This study primarily demonstrates the successful implementation of digital channelization processing on a domestic FPGA. The digital channelization workflow was modeled and simulated using MATLAB, and each module was implemented on the FPGA. An application test board was constructed based on the domestic FMQL45T900 FPGA, and the correctness of frequency measurement data was verified, confirming the feasibility of the digital channelization approach and improving processing efficiency. By employing digital signal processing techniques, broadband signals were decomposed into multiple narrowband channels, enabling parallel signal processing and frequency-selective analysis. The validation on a domestic FPGA demonstrated engineering adaptability and feasibility for domestic applications, effectively mitigating risks associated with foreign export restrictions and ensuring no delays in equipment development and maintenance. In the future, the increasing demand for processing multiple narrowband signals within high-bandwidth reception will impose higher efficiency requirements on digital signal processing. Imaginary number digital channelization processing remains a promising research direction.

Funding

This paper was supported by the National Natural Science Foundation of China (Grant No.: 62105176).

Data Availability

The data used to support the research findings are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] A. Samila, O. Hotra, and J. Majewski, "Implementation of the configuration structure of an integrated computational core of a pulsed NQR sensor based on FPGA," *Sensors*, vol. 21, no. 18, pp. 6029–6029, 2021. https://doi.org/10.3390/s21186029

[2] H. K. Mewada and J. Chaudhari, "Low computation digital down converter using polyphase IIR filter," *Circuit World*, vol. 45, no. 3, pp. 169–178, 2019. https://doi.org/10.1108/CW-02-2019-0015

[3] V. Thamizharasan and N. Kasthuri, "FPGA implementation of high performance digital FIR filter design using a hybrid adder and multiplier," *Int. J. Electron.*, vol. 110, no. 4, pp. 587–607, 2023. https://doi.org/10.1080/00207217.2022.2098387

[4] W. X. Zhang, X. L. Cui, Z. K. Zhao, and X. Q. Zhao, "Sub-nyquist sampling and gridless frequency estimation based on channelization," *IEEE Signal Process. Lett.*, vol. 30, pp. 1492–1496, 2023. https://doi.org/10.1109/LSP.2023.3324549

[5] D. Datta and H. S. Dutta, "High efficient half-band decimation filter on FPGA," *Trans. Indian Natl. Acad. Eng.*, vol. 8, no. 2, pp. 317–323, 2023. https://doi.org/10.1007/s41403-023-00392-0

[6] K. W. Kim, H. K. Kwon, and M. D. Kim, "Optimal equivalent-time sampling for periodic complex signals with digital down-conversion," *ETRI J.*, vol. 46, no. 2, pp. 238–249, 2024. https://doi.org/10.4218/etrij.2022-0426

[7] H. L. Zhang, Y. Z. Zhang, S. C. Guo, X. Du, N. Wang, J. Wang, X. C. Ye, H. Wu, J. Li, X. Pei, and Q. Meng, "An ultrawide bandwidth digital backend system based on PFB algorithm for QTT," *Astron. J.*, vol. 168, no. 5, p. 222, 2024. https://doi.org/10.3847/1538-3881/ad7fe0

[8] M. Zhang, H. L. Zhang, Y. Z. Zhang, J. Wang, S. C. Guo, and Q. Meng, "Research on channelization techniques of radio astronomical wideband signal with oversampled polyphase filter banks," *Res. Astron. Astrophys.*, vol. 23, no. 8, p. 085012, 2023. https://doi.org/10.1088/1674-4527/acd73b

[9] Y. X. Liu, Y. F. Guo, and S. B. Wu, "Frequency measurement of microwave signals in a wide frequency range based on an optical frequency comb and channelization method," *Appl. Opt.*, vol. 61, no. 13, pp. 3663–3670, 2022. https://doi.org/10.1364/AO.456507

[10] M. Huiskamp, M. S. O. Alink, B. Nauta, A. J. Annema, and H. S. Bindra, "Reconstructing aliased frequency spectra by using multiple sample rates," *IEEE Trans. Circuits Syst. I: Regul. Pap.*, vol. 69, no. 3, pp. 999–1012, 2021. https://doi.org/10.1109/TCSI.2021.3131273

[11] Z. Song, J. Yang, H. Zhang, and Y. Gao, "Approaching sub-nyquist boundary: Optimized compressed spectrum sensing based on multicoset sampler for multiband signal," *IEEE Trans. Signal Process.*, vol. 70, pp. 4225–4238, 2022. https://doi.org/10.1109/TSP.2022.3198186

[12] G. Huang, A. Ni, W. Lu, H. Peng, and J. Wang, "Parameters measurement of multiple exponentially damped sinusoids with sub-nyquist sampling," *IEEE Trans. Circuits Syst. II, Express Briefs*, vol. 68, no. 7, pp. 2710–2714, 2021. https://doi.org/10.1109/TCSII.2021.3055938

[13] B. S. Girish, S. H. Reddy, S. Sethi, K. S. Srivani, R. Abhishek, B. Ajithkumar, S. Bhattramakki, K. Buch, S. Chaudhuri, Y. Gupta, P. A. Kamini, S. Kudale, S. Madhavi, M. Muley, T. Prabu, A. Raghunathan, and G. J. Shelton, "Progression of digital-receiver architecture: From MWA to SKA1-Low, and beyond," *J. Astrophys. Astron.*, vol. 44, p. 28, 2023. https://doi.org/10.1007/s12036-023-09921-3

[14] A. Whipple, M. W. Ruzindana, M. C. Burnett, J. W. Kunzler, K. Lyman, B. D. Jeffs, and K. F. Warnick, "Wideband array signal processing with real-time adaptive interference mitigation," *Sensors*, vol. 23, no. 14, p. 6584, 2023. https://doi.org/10.3390/s23146584

[15] S. Gunda and R. S. Ernest Ravindran, "Design and simulation of low power consuming digital controlled oscillator in all digital phase locked loop," *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, pp. 3801–3806, 2019. http://doi.org/10.12785/ijcds

[16] A. Mohammadian and C. Tellambura, "RF impairments in wireless transceivers: Phase noise, CFO, and IQ imbalance–A survey." *IEEE Access*, vol. 9, pp. 111 718–111 791, 2021. http://doi.org/10.1109/ACCESS.2021.3101845

[17] K. Takahashi, R. Roberts, Z. Jiang, and B. Memarzadeh, "Statistical evaluation of signal-to-noise ratio and timing jitter in equivalent-time sampling signals," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–4, 2021. http://doi.org/10.1109/TIM.2021.3078003

[18] M. Yin and Y. Jiang, "Enhanced signal processing through FPGA-based digital downconversion via the CORDIC algorithm," *J. Ind. Intell.*, vol. 2, no. 1, pp. 14–30, 2024. https://doi.org/10.56578/jii020102

[19] D. Liang, A. W. Mohammad, C. Roeloffzen, Q. G. Tan, L. Li, X. J. Li, B. Shao, B. Zhang, X. K. Deng, F. T. Zheng, L. Wevers, R. Grootjans, P. Kapteijn, R. B. Timens, R. Heuvink, and S. Musa, "Chip-based microwave photonic payload repeater for high throughput satellites," *Laser Photonics Rev.*, vol. 18, no. 2, p. 2200952, 2024. https://doi.org/10.1002/lpor.202200952

[20] F. Teng, Y. Jiao, W. Yang, J. Yan, Z. Gao, and Z. Lu, “Multiphase parallel demodulation for remote sensing satellite data transmission–filter bank based on WOLA structure,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 9556–9565, 2022. https://doi.org/10.1109/JSTARS.2022.3201251

[21] T. Chen, L. Shi, and L. Guo, “Gridless direction of arrival estimation exploiting sparse linear array,” *IEEE Signal Process. Lett.*, vol. 27, pp. 1625–1629, 2020. https://doi.org/10.1109/LSP.2020.3021276