Journal of Industrial Intelligence
https://www.acadlore.com/journals/JII

Enhanced Signal Processing Through FPGA-Based Digital Downconversion via the CORDIC Algorithm

Ming Yin*, Yi Jiang

School of Information Science and Technology, Qingdao University of Science and Technology, 266100 Qingdao, China

* Correspondence: Ming Yin (LD2020052901@qust.edu.cn)

Received: 02-08-2024 **Revised:** 03-10-2024 **Accepted:** 03-19-2024

Citation: 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.

Abstract: To address the rate matching issue between high-bandwidth and high-sampling-rate analog-to-digital converters (ADCs) and low-bandwidth and low-sampling-rate baseband processors, the key technology of digital downconversion is introduced. This approach relocates the intermediate-frequency baseband signal to a vicinity of the baseband, laying a foundation for subsequent Digital Signal Processor (DSP) analysis and processing. In an innovative application of the Coordinate Rotation Digital Computer (CORDIC) algorithm for Numerically Controlled Oscillator (NCO) in a pipeline design, the phase differences of five parallel signals are measured, facilitating real-time parallel processing of the phase and amplitude relationships of multiple signals. The Field Programmable Gate Array (FPGA) design and implementation of the digital mixer module and filter bank for digital downconversion have been accomplished. A test board for the direction-finding application of five digital downconversion channels has been constructed, with the FMQL45T900 as its core. The correctness of the direction-finding data has been validated through practical application, demonstrating a significant improvement in power consumption compared to methods documented in other literature, thereby enhancing overall efficiency. The digital downconversion technology based on the CORDIC algorithm is applicable in various fields, including military communications, broadcasting, and radar navigation systems.

Keywords: Coordinate rotation digital computer algorithm; Digital downconversion; Field programmable gate array design; Pipeline architecture; Filters

1 Introduction

The concept of software-defined radio entails the processing of analog signals received by an ADC through software. However, due to the constraints imposed by the high sampling rates of ADCs, and the discrepancies in operational speeds dictated by the sampling theorem for DSPs, direct digital signal processing presents significant challenges. This has led to the adoption of intermediate-frequency reception technology in software-defined radio technologies. IF reception technology employs mixing to shift high-speed signals to lower speeds, subsequently allowing DSPs to carry out backend processing. Positioned between the ADC and the DSP, digital downconversion plays a pivotal role. It shifts signals around the intermediate frequency bandwidth closer to the baseband and applies extraction filtering. This reduces the sampling rate required for reception, preserves useful signals, and significantly alleviates the workload on DSPs [1]. Digital downconversion has been widely applied in fields such as military communications, broadcasting, television, and radar navigation systems, providing them with robust signal processing capabilities [2].

Datta et al. [3] proposed an efficient FPGA-based design for digital downconversion, specifically tailored for software-defined radio applications. The efficient application of digital filters on FPGA platforms was designed [4, 5]. Datta and Dutta [6] designed and implemented half-band (HB) filters within Digital Downconverters (DDCs). Kim et al. [7] applied digital downconversion technology for sampling equivalent time signals in complex periodic signals. Liu et al. [8] explored the application of digital downconversion technology at high sampling rates. Datta and Dutta [9] extended the use of digital downconversion technology to the field of wireless communications. Datta et al. [10] introduced a novel digital downconversion architecture, integrating a CORDIC processor for multi-channel finite impulse response (FIR), yielding favorable outputs. Sikka et al. [11] presented advanced synthesis implementations aimed at optimizing power consumption and area for DDCs. Bai et al. [12] realized the implementation of digital downconversion technology based on software-defined radio.

Despite the extensive application of digital downconversion algorithms, validation and expansion in specific projects remain unexplored. To overcome technological bottlenecks while minimizing manufacturing costs, a path of self-reliance and independent research and development is sought to achieve domestic substitution of core technologies. This study presents an FPGA-based design of a DDC utilizing the CORDIC algorithm. The design was implemented on a domestic FPGA chip, addressing the realization of phase differences across five channels, and achieving higher direction-finding accuracy compared to other literature. Digital downconversion technology plays a crucial role in both defense military applications and civilian communication domains [13].

2 Structural Composition of Digital Downconversion

From a working principle perspective, DDCs share similarities with their analog counterparts. Both digital and analog downconverters accomplish the reduction of intermediate-frequency signals and the extraction of baseband signals through the mixing operation, which involves multiplying the intermediate-frequency signal with orthogonal sine and cosine signals generated by a NCO [14]. In this process, the signals are subjected to extraction and filtering through filter processing. Despite the similar working principles, DDCs exhibit several distinct advantages [15]. Firstly, while analog downconversion is susceptible to the speed limitations of DSPs, the processing speed of digital downconversion is unrestricted. Secondly, the operational speed of analog downconversion determines the maximum data flow rate that can be processed, whereas DDCs can achieve high-speed, high-precision processing. Lastly, data accuracy and computational precision significantly impact the overall performance of DDCs [16]. Hence, the optimization and analytical design of DDCs are crucial, as they directly influence the stability and reliability of system performance. In summary, digital downconversion technology possesses considerable advantages, promising a broad spectrum of applications. Nonetheless, further research and optimization are required to enhance its performance and reliability.

The functionalities of digital downconversion are primarily divided into three parts:

(i) Frequency conversion: Digital mixers are employed to generate two orthogonal oscillating signals at high-speed ADCs, controlling the system's operation.

(ii) Low-pass (LP) filtering: This stage involves filtering out signals not belonging to the base frequency and extracting valuable signals. It is accomplished by a high-order FIR filter.

(iii) Sampling rate conversion: The sampling rate is reduced to facilitate subsequent processing by DSPs, thereby simplifying data handling.

The working principle is illustrated as follows, with the input signal denoted as:

$$f(n) = A\cos\left[w_c n + \varphi(n)\right]$$

The signal generated by the local oscillator is denoted as $\cos(w_c n)$ and $\sin(w_c n)$, resulting in:

$$y_I(n) = A/2\left\{\cos\left[2w_c n + \varphi(n)\right] + \cos[\varphi(n)]\right\}$$
$$y_Q(n) = A/2\left\{\sin\left[2w_c n + \varphi(n)\right] - \sin[\varphi(n)]\right\}$$

The above equations represent the In-phase (I) and Quadrature (Q) signals, which are then passed through a LP filter to obtain:

$$I'(n) = \frac{A}{2}\cos[\varphi(n)]$$
$$Q'(n) = -\frac{A}{2}\sin[\varphi(n)]$$

These are the baseband signals. However, at this stage, the signal rate remains high and does not meet the requirements for subsequent processing. A decimation filter is required to reduce the rate to satisfy backend processing requirements.

Given that the I and Q signals output by the DDC system need to undergo signal decoding, modulation style recognition, and signal parameter estimation for subsequent processing, the orthogonally decomposed baseband signals facilitate and enhance performance for these processes. Consequently, the typical structure of a DDC is illustrated in Figure 1.

When ADCs perform bandpass sampling on intermediate-frequency signals, a higher sampling rate may be necessitated. This elevated rate could hinder the normal operation of subsequent FIR filters. Therefore, in processing these signals, an initial substantial decimation is required through cascaded integrator-comb (CIC) filters and HB filters to reduce the data rate and meet the operational requirements of subsequent processing stages. Subsequently, shaping filtering is conducted by FIR filters.

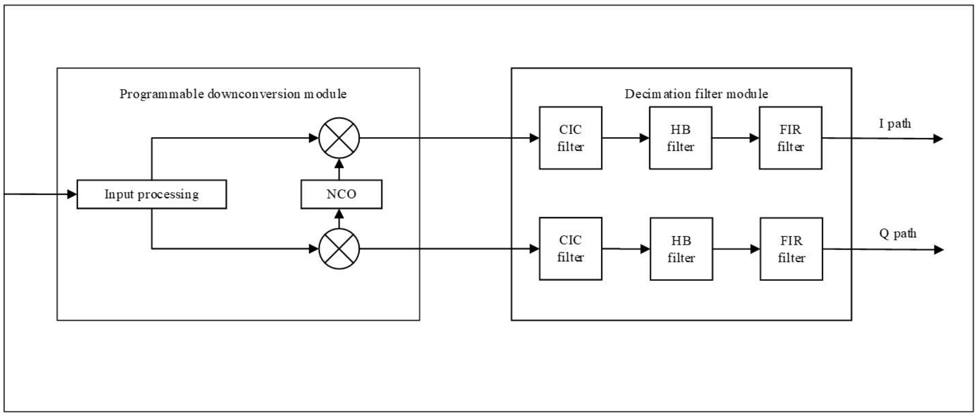


Figure 1. Block diagram of a DDC

The coefficients for both CIC and HB are set to one, necessitating only addition and subtraction operations without the need for multiplication, thereby simplifying implementation. Moreover, FPGA technology enables higher parallel processing speeds, making it suitable for the initial decimation in filter banks and for decimation processing with larger decimation factors.

However, due to their inherent structural characteristics, CIC filters exhibit poor spectral performance, necessitating cascading multiple CIC filters for improvement. With coefficients ranging from 1 to 32, a 3rd-order CIC filter can achieve a bandwidth of 67.3 dB. The structural feature of the HB filter, where half of its coefficients are zero, effectively halves the computational load. Thus, it serves as the subsequent stage of LP decimation filtering. The decimation factor of the HB filter is two, easily achieving a halving of the rate. After processing by CIC and HB filters, the baseband signal in the input is reduced from the original sampling frequency to a lower rate, facilitating easy handling by subsequent FIR filters. The FIR filter primarily shapes the signals decelerated by CIC and HB filters, eventually outputting the complete signal.

3 Design and Simulation of DDCs

3.1 Design of Digital Mixing Module

3.1.1 Programmable downconverter design

The orthogonal digital mixer (also known as the IQ mixer) is utilized for downconverting high-frequency signals (radio-frequency signals) to the intermediate-frequency range for processing. The principle involves dividing the input high-frequency signal into two paths, one path being multiplied (mixed) with a sine wave signal; the other path is multiplied with a sine wave signal after a 90-degree phase shift (cosine signal), resulting in two orthogonal signals. These two signals are referred to as the I and Q paths. The I and Q paths share the same frequency but differ by a 90-degree phase shift, thus enabling the representation of the original signal's amplitude and phase information, as shown by the equations.

$$I = A\cos[2\pi ft + \varphi]$$
$$Q = A\sin[2\pi ft + \varphi]$$

where, A denotes the amplitude of the original signal, f denotes the frequency of the original signal, t denotes time, and φ represents the phase difference of the original signal. By processing the I and Q path signals through digital signal processing, numerous functions can be achieved, such as demodulation, modulation, and spectrum analysis. Orthogonal digital mixing enables high-precision phase and amplitude adjustments in the digital domain, avoiding issues such as local oscillator drift and spurious responses found in traditional mixers.

The orthogonal digital mixer is one of the core technologies for implementing DDCs. This technology divides the high-frequency signal into sine and cosine paths, generating two orthogonal signals. The signal to be downconverted is then multiplied by the complex conjugate of the sine and cosine signals generated by the NCO, thereby achieving the downconversion of high-frequency signals to the intermediate-frequency range for processing. In this context, a NCO was utilized to generate sine and cosine waves.

In the field of digital communications, modems are indispensable, with the NCO serving as a core component of the modem. The sine and cosine signals generated by the NCO can be used for various modulation schemes. However, the controllability of high-frequency carrier signals poses a key challenge for high-speed digital communication systems. The advent of FPGAs offers a solution to this problem.

3.1.2 Implementation of NCO based on the CORDIC algorithm

The fundamental concept of the CORDIC algorithm is approached through a series of fixed, specially designed angles related to the base of operations, which are cumulatively added to approximate the desired rotation angle. Characterized by the use of shifting and addition/subtraction operations, it recursively computes the values of common functions, such as sine and cosine [17].

Within the xy coordinate plane, a point $(x1, y1)$ is rotated by an angle θ to a new position $(x2, y2)$. This relationship is represented by the following equations:

$$x_2 = x_1 \cos\theta - y_1 \sin\theta$$
$$y_2 = x_1 \sin\theta + y_1 \cos\theta$$

These equations can be expressed in matrix-vector form as shown:

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

By extracting the factor $\cos\theta$, the expression can also be written in the following form:

$$x_2 = x_1 \cos\theta - y_1 \sin\theta = \cos\theta\,(x_1 - y_1 \tan\theta)$$
$$y_2 = x_1 \sin\theta + y_1 \cos\theta = \cos\theta\,(y_1 + x_1 \tan\theta)$$

If the term $\cos\theta$ is eliminated, defining a pseudo-rotation equation:

$$\hat{x}_2 = x_1 - y_1 \tan\theta$$
$$\hat{y}_2 = y_1 + x_1 \tan\theta$$

The essence of the CORDIC method lies in the pseudo-rotation angle, where $\tan\theta = 2 - I$. Thus, the equation can be represented as:

$$\hat{x}_2 = x_1 - y_1 \tan\theta = x_1 - y_1 2^{-i}$$
$$\hat{y}_2 = y_1 + x_1 \tan\theta = y_1 + x_1 2^{-i}$$

The CORDIC algorithm transforms the process into an iterative method. By restricting the range of possible rotation angles, the rotation for any given angle can be achieved through a series of iterative small-angle rotations i, where multiplication by the tangent term is replaced by shift operations. The previously mentioned pseudo-rotation can now be expressed as (for each iteration) [18]:

$$x^{(i+1)} = x^{(i)} - d_i \left(2^{-i} y^{(i)}\right)$$
$$y^{(i+1)} = y^{(i)} + d_i \left(2^{-i} x^{(i)}\right)$$

Furthermore, a third equation is introduced, known as the angle accumulator, which is utilized to track the cumulative rotation angle throughout each iteration:

$$z^{(i+1)} = z^{(i)} - d_i \theta^{(i)}$$

where, $d_i = \pm 1$. The sign d_i serves as a decision operator, determining the direction of rotation.

$$d_i = \text{sign}\left(z^{(i)}\right)$$

After n iterations, the iterative values converge to the final value.

$$x^{(n)} = k_n \left(x^{(0)} \cos z^{(0)} - y^{(0)} \sin z^{(0)}\right)$$
$$y^{(n)} = k_n \left(y^{(0)} \cos z^{(0)} + x^{(0)} \sin z^{(0)}\right)$$
$$z^{(n)} = 0$$

The CORDIC algorithm, implemented using a pipeline structure, occupies more resources than an iterative structure but significantly increases the data throughput rate. The pipeline structure unfolds the iterative structure, allowing each of the n processing units to concurrently process the same iterative operation in parallel [19].

As depicted in Figure 2, the steps for implementing the CORDIC algorithm are as follows:

(i) An initial set of values, $x0$ and $y0$, is provided for the CORDIC algorithm implementation, with the input variable $z0$ being the angle variable.

(ii) Initially, X and Y are input into a shift register with a fixed number of shifts.

(iii) The result is then fed into an adder/subtractor.

(iv) The operation of the adder/subtractor, whether addition or subtraction, is determined based on the output of the angle accumulator, thus completing one iteration.

(v) The result of this iteration is passed as input to the next level of iterative computation, with the iterative process continuing sequentially.

(vi) Upon reaching the required number of iterations (in this case, seven), the result is outputted, yielding the desired outcome.

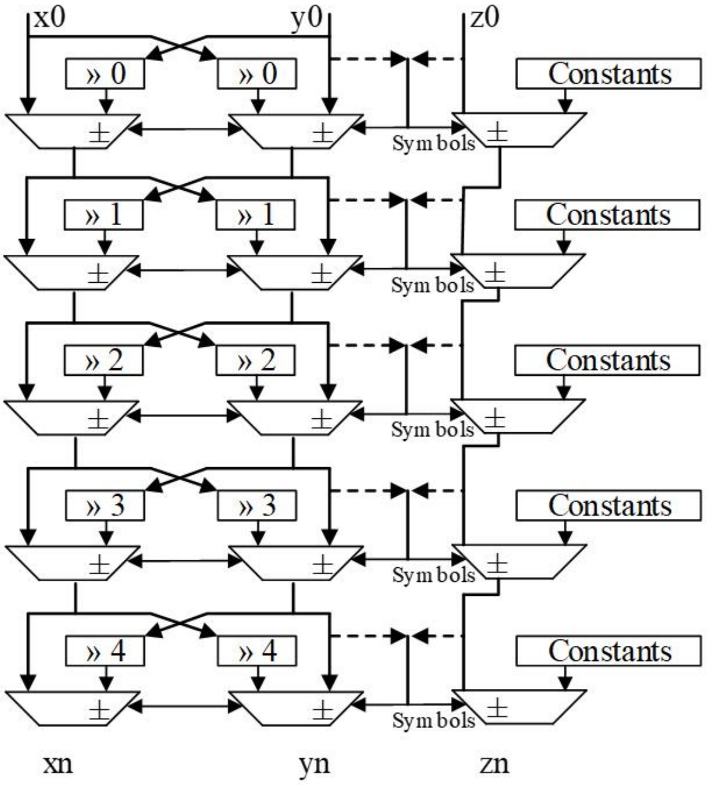


Figure 2. Implementation of the CORDIC algorithm

Therefore, the entire CORDIC processor implemented in FPGA comprises an internally interconnected array of adders/subtractors.

3.2 Design of Digital Decimation Filters

3.2.1 CIC filter design

(i) Single-stage CIC filter

The CIC filter is a type of digital filter composed of multiple integrators and comb filters in a cascading configuration. It is known for its efficiency, simplicity, and ease of hardware implementation, making it widely used in the field of digital signal processing. The basic structure of a CIC filter consists of a cascade of an integrator followed by a comb filter. The integrator, formed by an adder and a delay element, integrates the input signal to achieve filtering effects. The comb filter, consisting of multiple delay elements and an adder, samples and reconstructs the output signal from the integrator, thereby further achieving filtering effects [20].

The integration module comprises N ideal integrators, each acting as a single-pole Infinite Impulse Response (IIR) filter. The integrator can be viewed as an accumulator, summing and integrating the input signal to effectuate filtering. According to the Z-transform, the transfer function of the integrator can be expressed as:

$$H_I(z) = \frac{1}{1 - z^{-1}}$$

The comb filter is a symmetric FIR filter, with its corresponding transfer function as:

$$H_I(\mathrm{z}) = 1 - z^{-D}$$

The structure of a single-stage CIC filter is illustrated in Figure 3.

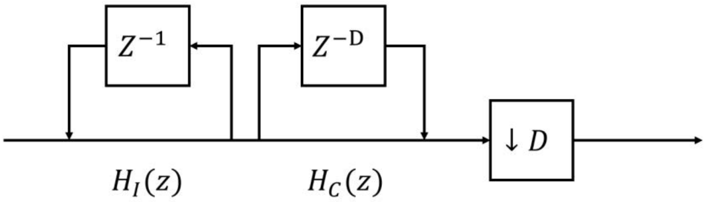


Figure 3. Structure of a single-stage CIC filter

In the figure, D represents the decimation factor. The single-stage CIC filter exhibits characteristics below. The impulse response of a single-stage CIC is:

$$h(n) = \begin{cases} 1, & 0 \leq n \leq D-1 \\ 0, & \text{else} \end{cases}$$

The frequency characteristics of a single-stage CIC filter are as follows:

$$H\left(e^{j\omega}\right) = H_I\left(e^{j\omega}\right) \cdot H_C\left(e^{j\omega}\right) = D \cdot Sa\left(\frac{\omega D}{2}\right) \cdot Sa^{-1}\left(\frac{\omega}{2}\right) \cdot e^{-j\omega\frac{D-1}{2}}$$

The transfer function of a single-stage CIC filter is presented:

$$H(z) = H_I(z) \cdot H_C(z) = \frac{1}{1-z^{-1}} \cdot \left(1-z^{-D}\right)$$

where, $H(z)$ represents the recursive form of an IIR filter. However, when D is not equal to 1 in practical situations, $H(z)$ can be simplified to:

$$H(z) = \sum_{n=0}^{D-1} z^{-n}$$

Although $H(z)$ is recursive, it can also be represented as an FIR filter. When a displacement of an FIR filter is equal, generally D-1 adders are required. However, in a CIC filter circuit, the same effect can be achieved with only one addition and one subtraction circuit. This is because the basic structure of a CIC filter includes multiple integrators and comb filters, enabling multi-stage filtering effects. This reduces the complexity and cost of the hardware.

However, the sidelobe level of a single-stage CIC filter is high, indicating poor stopband attenuation of the LP filter, making it difficult to meet the requirements of practical applications. A cascading method involving multiple CIC filters can be employed to reduce the sidelobe level. By cascading multiple CIC filters, the high-frequency components in the input signal can be further filtered out, lowering the sidelobe level and thus yielding a smoother filter output signal. This method aids in enhancing the performance and effectiveness of the entire filtering system.

(ii) Multi-stage CIC filters and their optimized efficient structures

A multi-stage CIC filter typically consists of a cascade of an integrator comb filter and an integrator filter. Their transformation expressions are as follows:

$$H_I(z) = \left(\frac{1}{1-z^{-1}}\right)^N$$

$$H_C(z) = \left(1-z^{-RM}\right)^N$$

Cascading these two expressions, the transformed expression for the transfer function becomes:

$$H(z) = \left(\frac{1-z^{-RM}}{1-z^{-1}}\right)^N$$

where, R represents the decimation factor of the decimation part, M is the differential delay of the comb part, typically 1 or 2, and N is the number of cascading stages. Illustrating with a three-stage CIC filter, after the Noble identity transformation, the structure of a three-stage CIC decimation filter is shown in Figure 4.

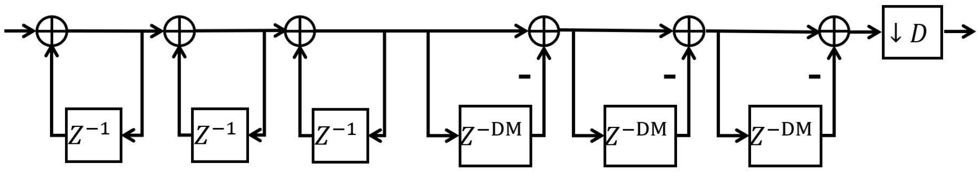


Figure 4. Equivalent transformation structure of a 3-stage CIC filter

The position of the comb filter and the integrator filter has little impact on system performance. However, theoretically, the position of the comb filter should be on the lower sampling rate side.

In practical applications, the decimator is usually placed before the comb filter, a configuration that allows for more efficient decimation. This arrangement contributes to the enhancement of the overall system's performance and efficiency. Similarly, with the intention of achieving more efficient operation, the integrator is positioned after the comb filter for interpolation. This ensures that the integrator operates at a higher sampling rate, while the interpolator functions at a lower sampling rate. Such a design approach aids in achieving efficient interpolation processing.

In summary, when implementing efficient decimation and interpolation, it is necessary to appropriately adjust the positions of the integrator and the comb filter, enabling them to operate at high and low sampling rates, respectively. This design strategy helps to improve the performance and efficiency of the entire signal processing system. The structure of the optimized CIC filter is shown in Figure 5, where the difference between interpolation and decimation lies in swapping the positions of the comb filter and the integrator.

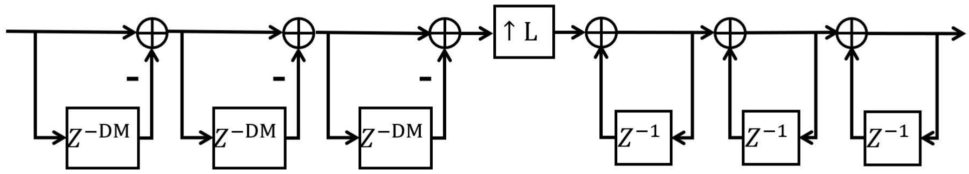


Figure 5. Equivalent transformation structure of an optimized efficient 3-stage CIC interpolation filter

3.2.2 The HB filter

The HB filter is utilized to filter signals within a frequency range bounded by two cutoff frequencies. Similar to LP and band-pass (BP) filters, it exhibits distinct transmission characteristics. The HB filter achieves its filtering effect by compressing signals within a frequency range into a narrower bandwidth. It has two cutoff frequencies, referred to as the lower and upper cutoff frequencies, with frequencies between these two being filtered out. Characteristics of the HB filter include:

$$H\left(e^{j\omega}\right) = 1 - H\left(e^{j(\pi-\omega)}\right)$$

$$H\left(e^{\frac{j\pi}{2}}\right) = 0.5$$

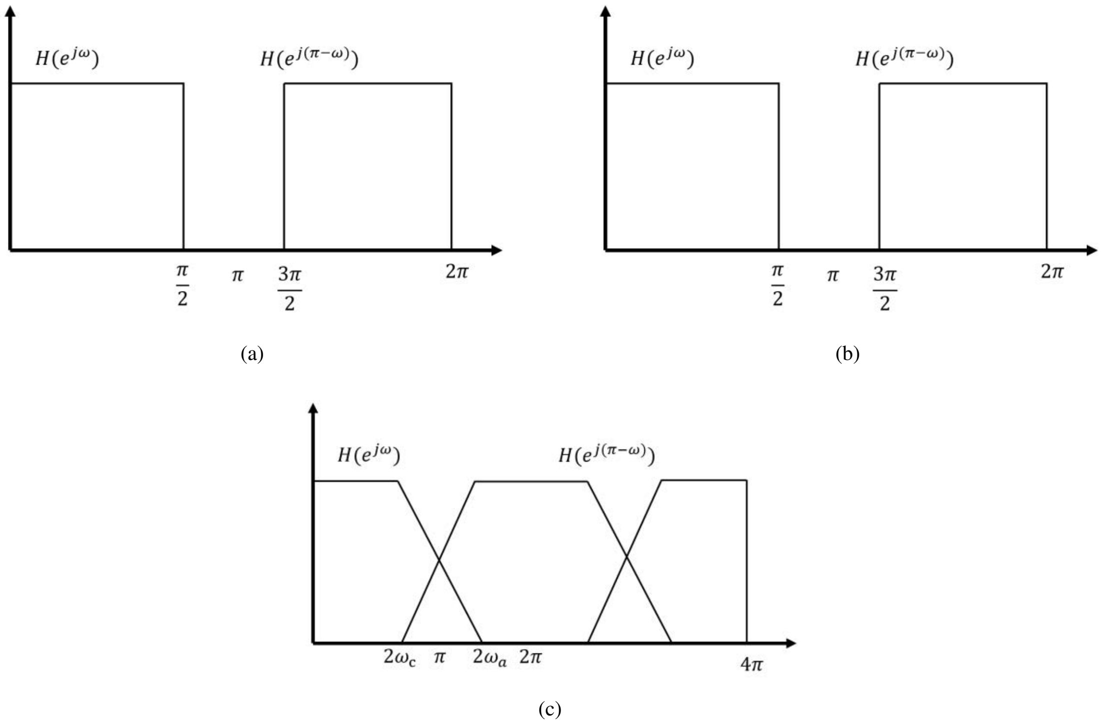
$$h(k) = \begin{cases} 1, & K = 0 \\ 0, & K = -2, 2, -4, 4 \ldots \end{cases}$$

In this study, aside from zero values, the even points of the HB filter are all zeros, meaning that the computational load is halved, thus enabling a very high data rate and excellent real-time performance [21].

As shown in Figure 6, the transition band of the HB filter in practical applications is not zero within the pi interval, thus not meeting the ideal conditions for aliasing-free decimation. However, provided the HB filter meets specific characteristics, the passband (0-w) post-decimation remains free of aliasing, implying that signals within the passband can be recovered after decimation. To achieve this goal, a meticulous design of filter parameters wc and wa is required based on the sampling rates and signal bandwidth before and after decimation.

Utilizing a single-stage HB filter, decimation or interpolation by a factor of 2 can be easily achieved. When the decimation rate is a power of 2 ($D = 2N$), decimation by a high factor of $D = 2N$ can be accomplished using N HB filters. This approach finds widespread application in digital signal processing.

For the design of filters that meet specific performance criteria, functions within MATLAB can be used to determine the filter coefficients. These coefficients are then converted into binary and inputted into a Verilog program for simulation and verification. Through this method, it is ensured that the designed filters possess desirable performance and meet the demands of practical applications [22].



(a)

(b)

(c)

Figure 6. Aliasing scenarios under the influence of a 2x decimation factor with a HB filter. (a) Ideal amplitude frequency response of halfband filter, (b) Amplitude frequency response of halfband filter, (c) Amplitude frequency response after decimation

3.2.3 The FIR filter

In the digital downconversion system, the FIR filter is situated at the final stage, with its primary role being the filtering and shaping of the signal output from the HB filter, without the need for decimation. Since the CIC and HB filters are primarily responsible for the decimation function to reduce the signal's frequency for subsequent DSP processing, and as the two preceding filters are already adequate for decimation, the main task of the FIR filter is thus focused on filtering and shaping. As depicted in Figure 7, the FIR filter employs a pipeline structure for implementation.

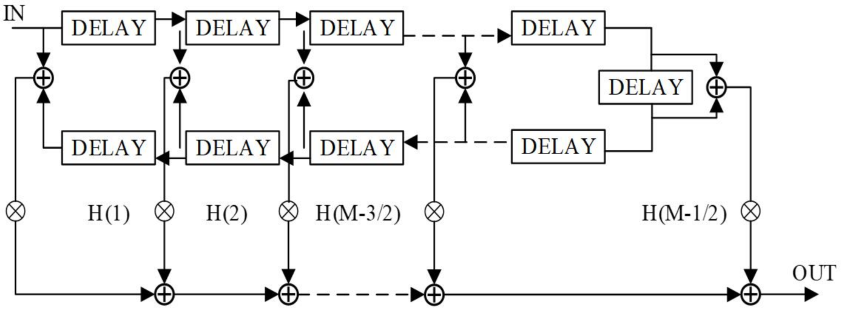


Figure 7. Pipeline implementation of the FIR filter

The pipeline design of the FIR filter involves inserting registers between sections (stages) to temporarily store intermediate data. This approach breaks down a large operation into several smaller operations, each with a shorter duration. These small operations can be executed in parallel, allowing data to flow through each stepin turn for processing, similar to an assembly line. From a holistic system perspective, this facilitates faster data input and output, thereby enhancing data throughput (increasing processing speed). The rate of the pipeline depends on the time consumed by each small operation. With the use of synchronous registers to segment the combinational logic, as long as the delay of each combinational logic is less than the clock cycle of the registers, the system's frequency is determined by the clock frequency of the system.

In FIR design, the pipeline structure is reasonably partitioned based on task characteristics. To facilitate better processing by subsequent DSP, the order of the preceding CIC filter can be increased to further reduce the data rate [23].

3.3 Simulation Based on MATLAB

The DDC, based on MATLAB, primarily comprises three components: the digital mixer, the NCO, and the filter. The analysis and simulation are focused around these three parts. Key parameters used in this design include an input sampling frequency of 170MHz, a modulated input signal carrier frequency of 384MHz, and a sampling bandwidth of 20MHz, as illustrated in Figure 8.

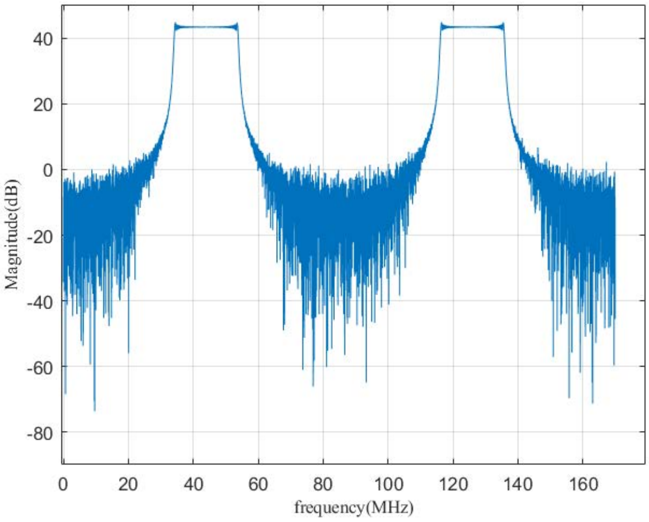


Figure 8. Frequency spectrum of the input signal. The spectrum of the BP signal sequence with a carrier frequency of 384MHz and a useful signal bandwidth of 20MHz

(i) The digital signal sequence generated by ADCs is represented by $xs = \cos\left(2^{*}pi^{*}\left(f0^{*}t + 0.5^{*}u^{*}t.^{\wedge}2\right)\right)$. This represents the output of the modulated input signal after BP sampling at fsamp=170MHz, equivalent to the digital signal sequence after the analog-to-digital conversion.

(ii) The NCO module utilizes the CORDIC algorithm for generation. It generates a cos numerically controlled local oscillator (I path) and a sin numerically controlled local oscillator (Q path) with a time interval identical to that of the ADC's sampling interval. The signal spectrum is shown in Figure 9.

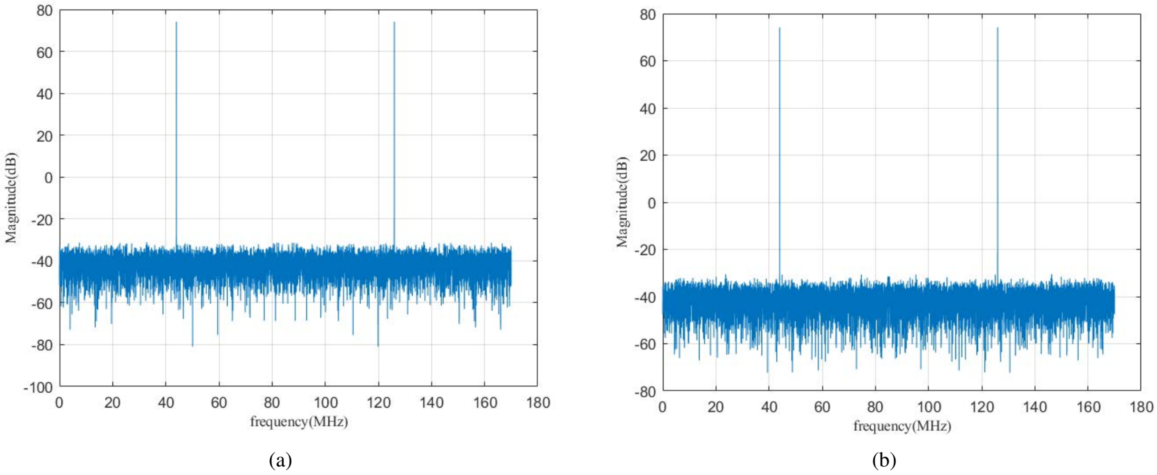


Figure 9. Spectrum generated by the NCO module. (a)The I-path signal spectrum with added Gaussian white noise, (b)The Q-path signal spectrum with added Gaussian white noise

(iii) The filter module

The basic principles and algorithms of the filter module have been described previously, and thus will not be elaborated upon further here.

For the HB filter, MATLAB's filter design tool can be utilized to calculate its coefficients, which can then be implemented in simulation software to generate the relevant filter.

After running the entire MATLAB program, the results are as depicted in Figures 10(a) and 10(b). The comparison of the two figures demonstrates that the initial input signal, after being processed by the entire system, successfully outputs a low-frequency signal, verifying the system's feasibility. This provides a theoretical basis for the subsequent design and implementation of digital downconversion based on FPGA.

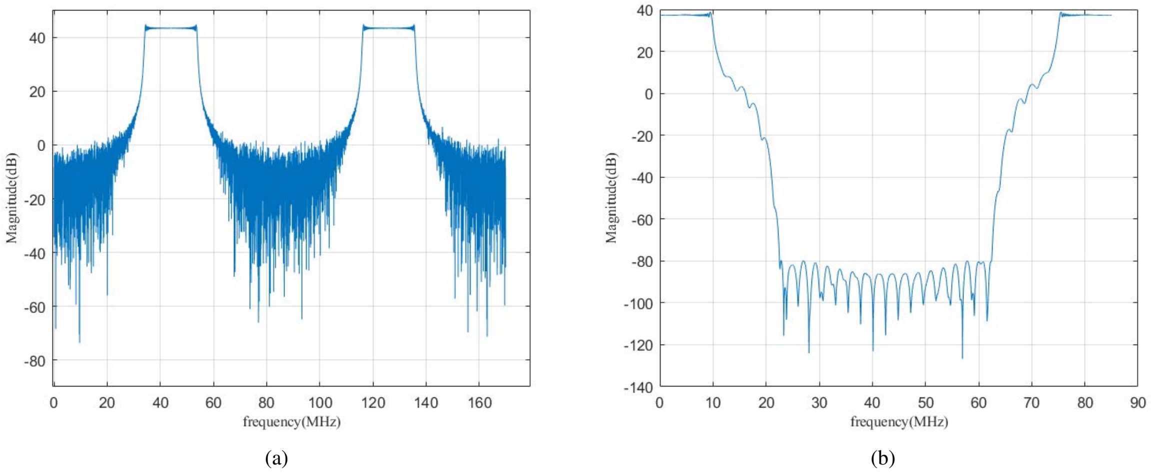


Figure 10. Frequency spectrum of input and output signals for the digital downconversion. (a) The spectrum of the BP signal sequence with a carrier frequency of 384MHz and a useful signal bandwidth of 20MHz, (b) The spectrum of the original A/D sampled signal sequence after downconversion and decimation sampling

In the FPGA implementation phase of this design, when a 0.5MHz signal is input into the digital downconversion system for processing, and the data file simulated through MODELSIM is invoked, MATLAB is used to plot the frequency spectrum of the input signal, as shown in Figure 11.

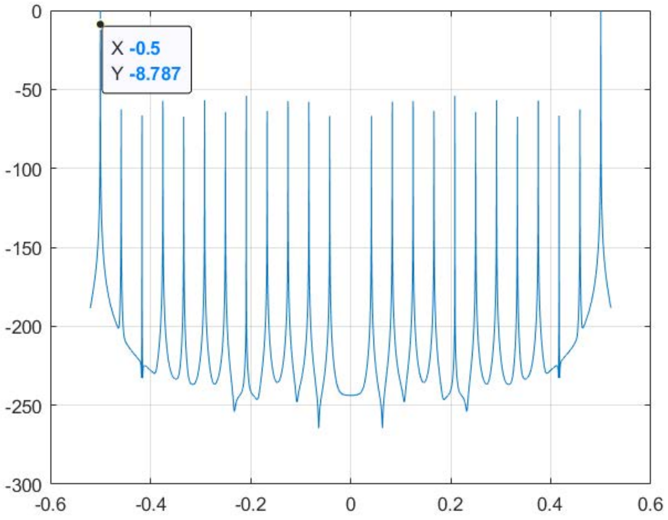


Figure 11. Spectrum of the input signal

Based on the theory mentioned above, the signal frequency generated by the NCO is 0.4MHz. With the DDC input signal at 0.5MHz, the processed output signal's I-path frequency is 0.1MHz.

The experimental results, as shown in Figure 12, confirm that the DDC output signal is indeed 0.1MHz, achieving the anticipated downconversion result and verifying the correctness of this design.

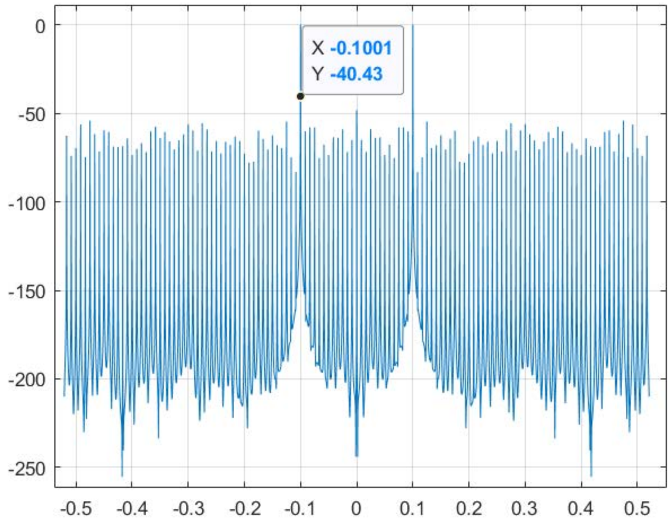


Figure 12. Spectrum of the output signal

4 FPGA Implementation of the DDC

4.1 NCO Design Using the CORDIC Algorithm

The CORDIC algorithm is employed to generate the carrier signal. As illustrated in Figure 13, the output from the initial phase accumulation part has a width of 24 bits, representing the phase within the range $[0, 2\pi]$. The iterative process then commences. In the first iteration, the corresponding phase angle undergoes preprocessing, and the interval is converted to $[-\pi, \pi]$, stored in register variables. The iteration continues, converting to the range $[-\pi/2, \pi/2]$, with subsequent iterations further refining the process and executing quadrant decision conversion to obtain the correct amplitude. The phase accuracy is achieved through iterations of the CORDIC algorithm, meaning that after i iterations, the phase accuracy obtained equals $\arctan\left(2^{-i}\right)$, leading to the calculation of the value of i. Requirements are satisfied when $i \geq 21$. Thus, employing a pipeline structure, the CORDIC algorithm iterates 21 times for implementation, with multi-level module instantiation carried out in Verilog.

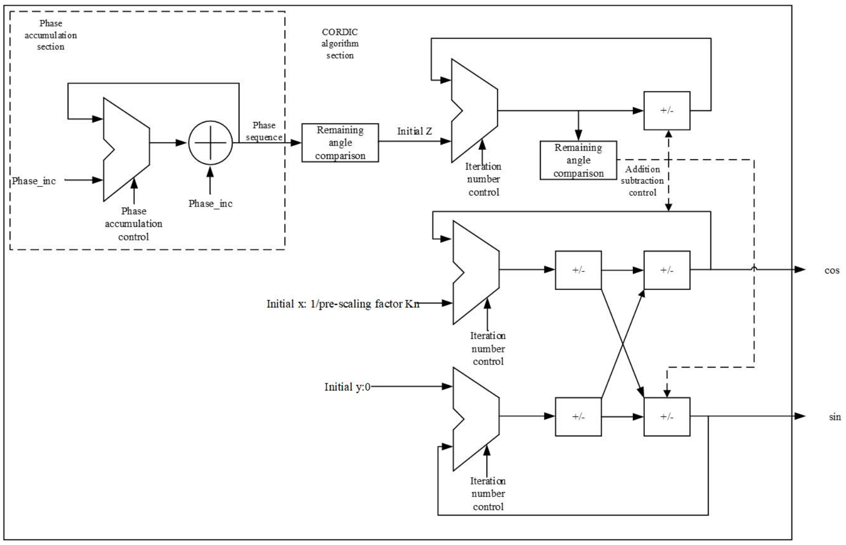
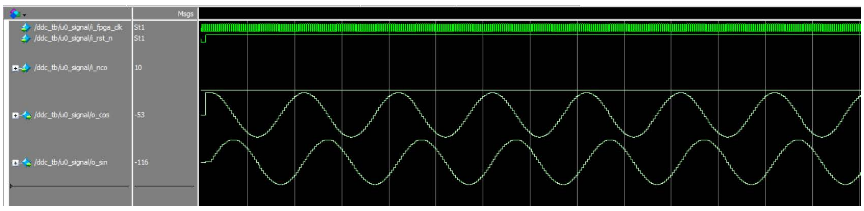


Figure 13. Pipeline-based CORDIC algorithm design

The implementation of the CORDIC algorithm through pipeline design has facilitated significant data throughput, alleviating the timing pressure of data operations and overall enhancing the operational speed of the NCO. As shown in Figure 14, it can be observed that the module generates sine and cosine signals that meet the design requirements.



```
/ddc_tb/u0_signal/i_fpga_clk    St1
/ddc_tb/u0_signal/i_rst_n       St1
/ddc_tb/u0_signal/i_nco         10
/ddc_tb/u0_signal/o_cos         -53
/ddc_tb/u0_signal/o_sin         -116
```

Figure 14. Simulation verification of the NCO module

4.2 Design of the Digital Mixer

The digital mixer, also known as a multiplier, is commonly used to multiply the sampled intermediate-frequency signal with the orthogonal signals generated by the NCO. Given that the sampling rate matches the data rate, the clock frequency of the mixer must also align with the sampling frequency.

The Register Transfer Level (RTL) schematic of the digital mixer module designed based on the aforementioned requirements is shown in Figure 15.

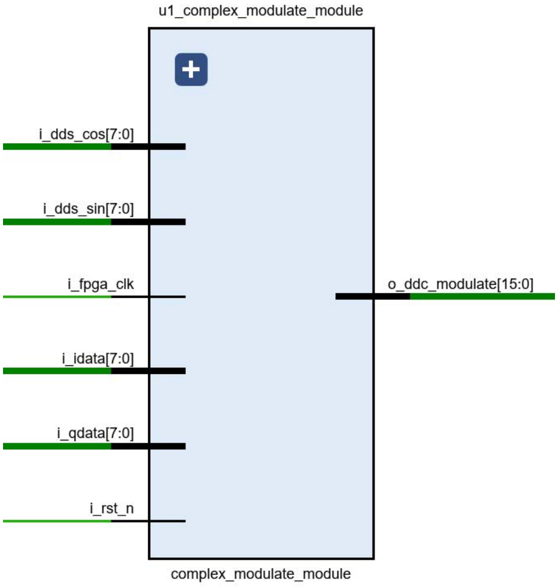
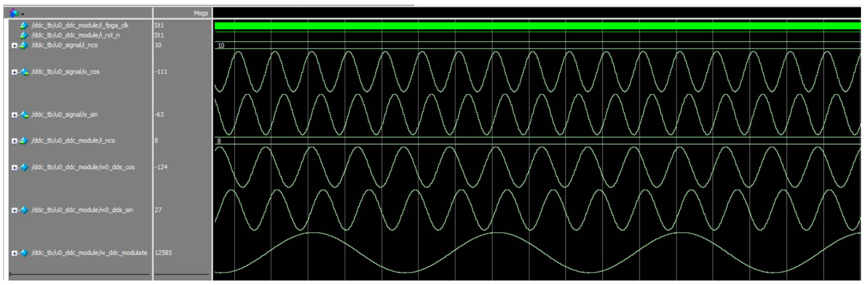


Figure 15. RTL schematic of the mixer module

The diagram illustrates two signals, *i_idata* and *i_qdata*, processed after analog-to-digital conversion; the two sine and cosine signals are generated by the NCO. These two sets of signals are mixed by the mixer and then output through *o_ddc_modulate*.

To verify the correctness of this design, a corresponding testbench was written, and the module was simulated using MODELSIM software. The simulation results are displayed in Figure 16.



```
/ddc_tb/u0_ddc_module/i_fpga_clk        St1
/ddc_tb/u0_ddc_module/i_rst_n           St1
/ddc_tb/u0_signal/i_nco                 10
/ddc_tb/u0_signal/o_cos                 -111
/ddc_tb/u0_signal/o_sin                 -63
/ddc_tb/u0_ddc_module/i_nco             8
/ddc_tb/u0_ddc_module/w0_dds_cos        -124
/ddc_tb/u0_ddc_module/w0_dds_sin        27
/ddc_tb/u0_ddc_module/w_ddc_modulate    12585
```

Figure 16. Simulation verification of the mixer module

The signal generated after processing by the mixer module meets the design requirements, confirming the functionality of the module is correct.

4.3 Design of the Filtering Module

The filter module comprises CIC, HB and FIR filters. The CIC and HB filters are primarily responsible for the decimation function to reduce the signal's frequency for subsequent DSP processing. Since the two preceding filters adequately perform the decimation function, the FIR filter's main role is focused on filtering and shaping tasks.

The CIC filter is a type of digital filter frequently used in digital signal processing as a decimation filter. Comprising cascaded integrators and combiners, the CIC filter enables efficient decimation filtering, characterized by its simple structure and low latency. The function of the CIC filter is to reduce the data rate to facilitate processing by subsequent DSP devices. The FIR filter's primary task is to filter and shape the signal decelerated by the preceding filters, outputting the complete signal, generally implemented using the Distributed Algorithm (DA) algorithm.

Described using the Verilog hardware description language and compiled in Vivado, the RTL schematic of the decimation and shaping filter module is illustrated as follows.

In the design, *i_cic_data* represents the mixed signal input, which, after decimation, frequency reduction, and shaping filtering, is output through *o_cic_data*, with *o_cic_fp* serving as the module completion flag. To verify the correctness of this design, a testbench for this module was written and simulated, with results shown in Figure 17.

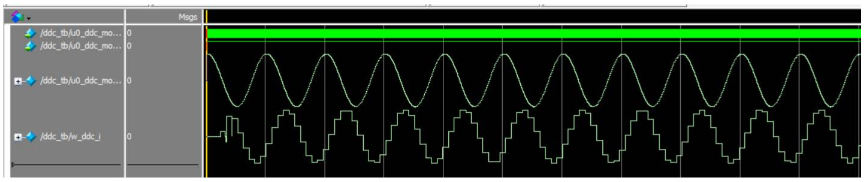


Figure 17. Simulation verification of the decimation and shaping filter module

The signal output by the module successfully completes the frequency reduction, verifying the functionality's correctness.

4.4 Optimized Simulation of the DDC

Described using the Verilog hardware description language and compiled in Vivado, the RTL schematic of the decimation and shaping filter module is illustrated. In this schematic, *i_cic_data* represents the mixed signal input, which, after decimation, frequency reduction, and shaping filtering, is output through *o_cic_data*. The *o_cic_fp* serves as the module completion flag.

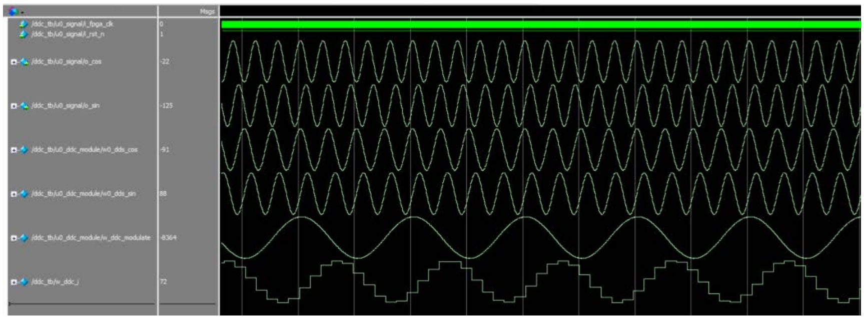

Msgs
/ddc_tb/u0_signal/i_fpga_clk 0
/ddc_tb/u0_signal/i_rst_n 1
/ddc_tb/u0_signal/o_cos -22
/ddc_tb/u0_signal/o_sin -125
/ddc_tb/u0_ddc_module/w0_dds_cos -91
/ddc_tb/u0_ddc_module/w0_dds_sin 88
/ddc_tb/u0_ddc_module/w_ddc_modulate -8364
/ddc_tb/w_ddc_i 72

Figure 18. Implementation of the DDC

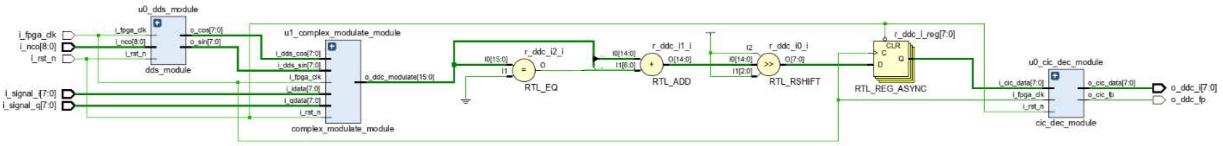


Figure 19. RTL schematic of the DDC

Following the implementation and verification of individual modules, which demonstrated the feasibility of each component, the entire digital downconversion system's implementation and verification could be carried out. By connecting the various modules of the DDC and writing the system's testbench, the results obtained after compiling and simulating in Vivado software are shown in Figure 18.

The frequency of the output waveform, obtained after processing by the entire digital downconversion system, successfully decreases, proving the system's correctness and feasibility. The RTL schematic of the DDC is shown in Figure 19.

5 Test Results

Testing of the digital downconversion was conducted on the domestic Fudan Microelectronics FMQL45T900 FPGA test board (Figure 20). As shown in Figure 21, five ADC channels were set up, collecting five intermediate frequency signals through the acquisition of five ADCs. A single RF signal generated by an RF generator was divided into five parallel signals through a power divider, input into five ADCs. Within the test board, analog-to-digital sampling, digital downconversion, and Fast Fourier Transformation (FFT) were completed, allowing for the extraction of corresponding phases; this setup enables the input of five intermediate frequency signals to parallelly complete analog-to-digital sampling, digital downconversion, and parameter extraction processing. The domestic Fudan Microelectronics FQL45 series FPGA is fully compatible with the American Xilinx Company's ZYNQ 7045 series. The main parameters of the ADC sub-board are as follows:

(i) Channel number: Five parallel channels;

(ii) ADC resolution: Low-speed channel ADC resolution is 16 bits;

(iii) Maximum sampling rate: The maximum sampling rate of the low-speed channel ADC is 250Msps;

(iv) Acquisition clock: Supports internal and external clocks;

(v) External trigger: Features an external trigger interface;

(vi) Synchronization accuracy: Better than 0.03ns;

(vii) Spurious-free dynamic range: Better than 85dBFS (low-speed channel);

(viii) Signal-to-noise ratio: Better than 70dBFS (low-speed channel).

After receiving analog-to-digital data through five ADC channels on the FPGA development board, the system processes this data through five digital downconversion channels for broadband data processing. It then performs parallel pipeline FFT processing on the five IQ data channels, uses calibration source data for channel correction after phase difference calculations, and carries out correlation operations to output the highest correlation value as azimuth information, thus obtaining the corresponding direction-finding results. Below is the validation of the DDC using the Integrated Logic Analyzer (ILA) in Vivado during board-level verification. The ILA captures trigger conditions such as the issued start command, the FPGA's control command for the RF module *cdf_spi_start*, the FPGA's phase difference calculation start signal *start_source*, the FPGA's correlation matching signal *start_match*, the direction-finding result output signal *cdf_pcie_wea*, and *cdf_rdy_busy* indicating the busy state of the entire direction-finding process, active high. The FPGA verification process is analyzed and monitored, with Figure 22 showing the output of the direction-finding results, receiving corresponding direction-finding information in the ILA.

The test system built on the test board, primarily comprising the domestically produced Fudan Microelectronics FMQL45T900, was verified by sampling the intermediate frequency signal, moving it near the baseband signal, and measuring the amplitude through FFT, as well as determining the phase through the direction-finding algorithm, thereby verifying the correct operation of the DDC.

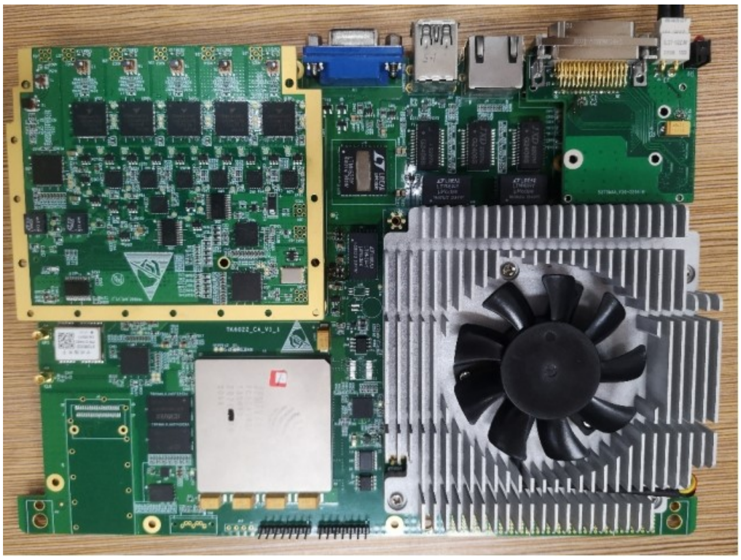


Figure 20. The test board

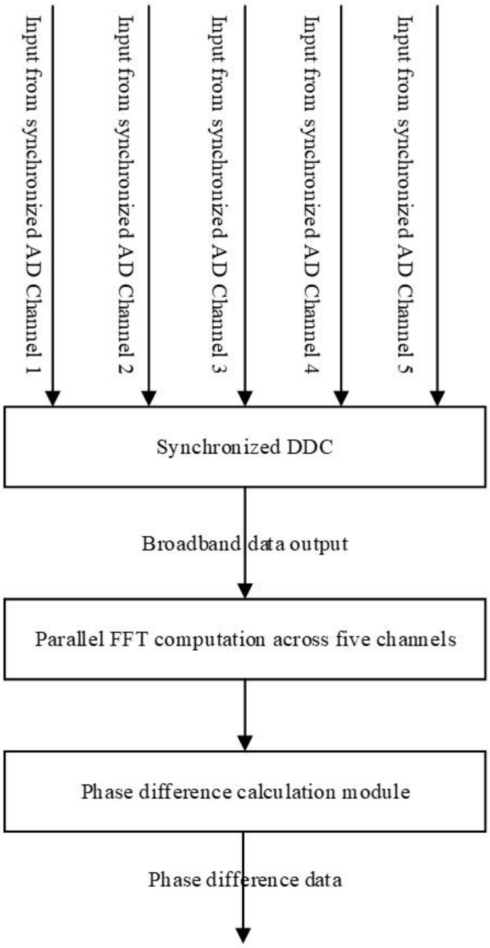
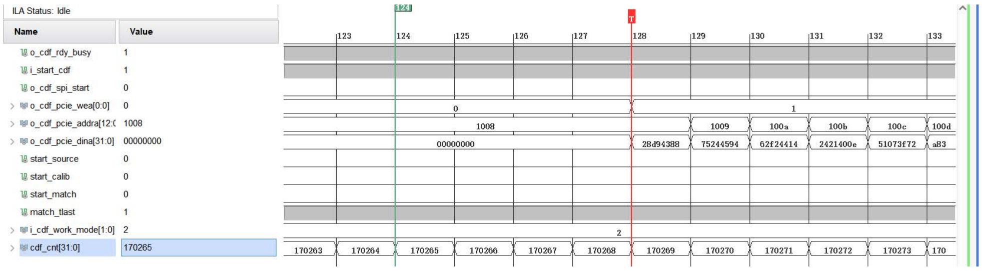

Input from synchronized AD Channel 1
Input from synchronized AD Channel 2
Input from synchronized AD Channel 3
Input from synchronized AD Channel 4
Input from synchronized AD Channel 5
Synchronized DDC
Broadband data output
Parallel FFT computation across five channels
Phase difference calculation module
Phase difference data

Figure 21. Flowchart of a direction-finding process using digital downconversion


ILA Status: Idle

Name	Value
o_cdf_rdy_busy	1
i_start_cdf	1
o_cdf_spi_start	0
o_cdf_pcie_wea[0:0]	0
o_cdf_pcie_addra[12:(	1008
o_cdf_pcie_dina[31:0]	00000000
start_source	0
start_calib	0
start_match	0
match_tlast	1
i_cdf_work_mode[1:0]	2
cdf_cnt[31:0]	170265

Figure 22. Diagram of the test results

The solution proposed in this study was implemented in the FMQL45T900, with the logic resources it occupies outlined in Table 1.

Table 1. Logic resources utilized by the method described in this study

Parameter	CORDIC Algorithm	DDC	DSP Block
Slice registers	148	380	40
Slices	86	78	10
Slice LUT	160	96	40
LUT flip-flop pairs	320	319	42
Bonded IOB	55	30	88

A comparison of the area, frequency, and power consumption between the method proposed in this study and the method described by Datta et al. [10] is presented in Table 2.

Table 2. Comparison of DDC architecture performance

Key Parameter	Datta et al. [10]	Method Proposed in This Study
Slice registers (area)	1882	1038
Frequency (MHz)	472.4	526
Power	1022	961

Table 2 clearly demonstrates that the method proposed reduces the number of registers compared to the work described by Datta et al. [10]. The reduction in registers enhances system performance and decreases power consumption. The operating frequency has been increased to 526MHz, further improving performance. Moreover, it can be effectively implemented on an FPGA.

6 Conclusions

The main findings of this study include the implementation of a pipeline design for the NCO using the CORDIC algorithm and the system design and modeling simulation of the entire DDC based on MATLAB. The FPGA design and implementation of the digital mixer module and filter group for digital downconversion were completed. A test board for the direction-finding application of five digital downconversion channels was constructed using the domestically produced FMQL45T900 as the core. The correctness of the direction-finding data was obtained, verifying the feasibility of the DDC and saving a significant amount of logic resources. The digital downconversion technology moves the intermediate-frequency baseband signal closer to the baseband, addressing the rate matching issue between high-bandwidth and high-sampling-rate ADCs and low-bandwidth and low-sampling-rate baseband processors. It was validated on a domestically produced FPGA, demonstrating high engineering adaptability and feasibility for domestic applications. In the future, with multiple narrowband signals needing processing within the receiving bandwidth, and high requirements for processing effectiveness, variable-speed digital downconversion will be a research direction.

Funding

This study was supported by the National Natural Science Foundation project (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, p. 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] D. Datta, P. Mitra, and H. S. Dutta, “Fpga implementation of high performance digital down converter for software defined radio,” *Microsyst. Technol.*, vol. 28, pp. 533–542, 2022. https://doi.org/10.1007/s00542-019-04579-w

[4] 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, 2022. https://doi.org/10.1080/00207217.2022.2098387

[5] Q. Jing, Y. Li, and J. Tong, “Performance analysis of multi-rate signal processing digital filters on FPGA,” *EURASIP J. Wireless Commun. Networking*, vol. 2019, no. 1, pp. 1–9, 2019. https://doi.org/10.1186/s13638-019-1349-9

[6] D. Datta and H. S. Dutta, “High efficient half-band decimation filter on FPGA,” *Trans. Indian Natl. Acad. Eng.*, vol. 8, pp. 317–323, 2023. https://doi.org/10.1007/s41403-023-00392-0

[7] K. W. Kim, H. K. Kwon, and M. D. Kim, “Optimal equivalenttime sampling for periodic complex signals with digital downconversion,” *ETRI J.*, 2023. https://doi.org/10.4218/etrij.2022-0426

[8] X. Liu, X. Yan, Z. Wang, and Q. Deng, “Design and FPGA implementation of a reconfigurable digital down converter for wideband applications,” *IEEE Trans. VLSI Syst.*, vol. 25, no. 12, pp. 3548–3552, 2017. https://doi.org/10.1109/tvlsi.2017.2748603

[9] D. Datta and H. S. Dutta, “Implementation of polyphase digital down converter for wireless applications,” *Microprocess. Microsyst.*, vol. 100, p. 104850, 2023. https://doi.org/10.1016/j.micpro.2023.104850

[10] D. Datta, P. Mitra, and H. S. Dutta, “Fpga implementation of high performance digital down converter for software defined radio,” *Microsyst. Technol.*, vol. 28, no. 2, pp. 533–542, 2019. https://doi.org/10.1007/s00542-019-04579-w

[11] P. Sikka, A. R. Abhijit, and C. Shekhar, “Power-and area-optimized high-level synthesis implementation of a digital down converter for software-defined radio applications,” *Circuits Syst. Signal Process.*, vol. 40, no. 6, pp. 2883–2894, 2020. https://doi.org/10.1007/s00034-020-01601-9

[12] F. M. Bai, X. L. Lv, and F. Yan, “Programming design of Digital-Down Converter based on software radio,” in *2010 Second Pacific-Asia Conference on Circuits, Communications and System, Beijing, China*, 2010, pp. 351–354. https://doi.org/10.1109/PACCS.2010.5626883

[13] S. Gunda and E. R. R. S. , “Design and simulation of low power consuming digital controlled oscillator in all digital phase locked loop,” *Int. J. Innovative Technol. Exploring Eng.*, vol. 8, no. 12, pp. 3801–3806, 2019. https://doi.org/10.35940/ijitee.L3830.1081219

[14] 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. https://doi.org/10.1109/access.2021.3101845

[15] 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. https://doi.org/10.1109/tim.2021.3078003

[16] T. S. Rappaport, Y. Xing, O. Kanhere, S. Ju, A. Madanayake, S. Mandal, A. Alkhateeb, and G. C. Trichopoulos, “Wireless communications and applications above 100 GHz: Opportunities and challenges for 6g and beyond,” *IEEE Access*, vol. 7, pp. 78 729–78 757, 2019. https://doi.org/10.1109/access.2019.2921522

[17] A. Changela, M. Zaveri, and D. Verma, “A comparative study on CORDIC algorithms and applications,” *J. Circuits Syst. Comput.*, vol. 32, no. 5, 2022. https://doi.org/10.1142/s0218126623300027

[18] A. Changela, M. Zaveri, and D. Verma, “Mixed-radix, virtually scaling-free CORDIC algorithm based rotator for DSP applications,” *Integration*, vol. 78, pp. 70–83, 2021. https://doi.org/10.1016/j.vlsi.2021.01.005

[19] F. Lyu, X. Xu, Y. Wang, Y. Luo, Y. Wang, and H. Pan, “Ultralow-latency VLSI architecture based on a linear approximation method for computing Nth roots of floating-point numbers,” *IEEE Trans. Circuits Syst. I*, vol. 68, no. 2, pp. 715–727, 2021. https://doi.org/10.1109/tcsi.2020.3038417

[20] H. Mahdavi and S. Timarchi, “Improving architectures of binary signed-digit CORDIC with generic/specific initial angles,” *IEEE Trans. Circuits Syst. I*, vol. 67, no. 7, pp. 2297–2304, 2020. https://doi.org/10.1109/tcsi.2020.2978765

[21] A. Changela, M. Zaveri, and D. Verma, “FPGA implementation of high-performance, resource-efficient Radix-16 CORDIC rotator based FFT algorithm,” *Integration*, vol. 73, pp. 89–100, 2020. https://doi.org/10.1016/j.vlsi.2020.03.008

[22] S. Mopuri and A. Acharyya, “Configurable rotation matrix of hyperbolic CORDIC for any logarithm and its inverse computation,” *Circuits Syst. Signal Process.*, vol. 39, pp. 2551–2573, 2020. https://doi.org/10.1007/s00034-019-01277-w

[23] Y. Wang, Y. Luo, Z. Wang, Q. Shen, and H. Pan, “GH CORDIC-based architecture for computing Nth root of single-precision floating-point number,” *IEEE Trans. VLSI Syst.*, vol. 28, no. 4, pp. 864–875, 2020. https://doi.org/10.1109/tvlsi.2019.2959847