

Research Article

FPGA-Based Implementation of All-Digital QPSK Carrier Recovery Loop Combining Costas Loop and Maximum Likelihood Frequency Estimator

Kaiyu Wang, Zhiming Song, Xianwei Qi, Qingxin Yan, and Zhenan Tang

*Dalian Institute of Semiconductor Technology and School of Electronic Science and Technology,
Dalian University of Technology, Dalian 116023, China*

Correspondence should be addressed to Zhenan Tang; tangza@dlut.edu.cn

Received 27 June 2014; Revised 3 August 2014; Accepted 6 August 2014; Published 31 August 2014

Academic Editor: Nadia Nedjah

Copyright © 2014 Kaiyu Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents an efficient all digital carrier recovery loop (ADCRL) for quadrature phase shift keying (QPSK). The ADCRL combines classic closed-loop carrier recovery circuit, all digital Costas loop (ADCOL), with frequency feedward loop, maximum likelihood frequency estimator (MLFE) so as to make the best use of the advantages of the two types of carrier recovery loops and obtain a more robust performance in the procedure of carrier recovery. Besides, considering that, for MLFE, the accurate estimation of frequency offset is associated with the linear characteristic of its frequency discriminator (FD), the Coordinate Rotation Digital Computer (CORDIC) algorithm is introduced into the FD based on MLFE to unwrap linearly phase difference. The frequency offset contained within the phase difference unwrapped is estimated by the MLFE implemented just using some shifter and multiply-accumulate units to assist the ADCOL to lock quickly and precisely. The joint simulation results of ModelSim and MATLAB show that the performances of the proposed ADCRL in locked-in time and range are superior to those of the ADCOL. On the other hand, a systematic design procedure based on FPGA for the proposed ADCRL is also presented.

1. Introduction

Along with the continuous development of the technologies of field programmable logic gate array (FPGA) and digital signal processing, the FPGAs in possession of large capacity and low power dissipation make it possible to realize a true software defined radio and integrate a whole digital communication system into the chips in order to reconfigure flexibly the continuously evolving communication protocols and minimize the volume of spacecraft. A typical example applying the software defined radio (SDR) based on FPGA to deep space communication is the National Aeronautic and Space Administration (NASA) Electra radio, in which the baseband processing is entirely implemented in a FPGA. Virtually any channel code, modulation, and data rate may be accommodated via suitable reprogramming of this SDR [1].

In communication system, the style of modulation and demodulation plays an important role and directly influences the performance of the system. However, in deep space communication, both power efficiency and bandwidth efficiency

of a communication system should be simultaneously considered. Therefore, the modulation and demodulation method of QPSK have been widely used into deep space communication. With respect to QPSK demodulator, there are two different solutions to demodulate. They are noncoherent demodulation and coherent demodulation, respectively. Compared with noncoherent demodulation, coherent demodulation can be implemented in a simpler structure so as to save the logic resources of FPGA. Hence, this paper takes the QPSK coherent demodulator as the object of our study.

In QPSK coherent demodulator, a special phase-locked loop (PLL), namely, Costas loop, is used to synchronize the locally generated carrier with the carrier contained in the received input buried by external noise. It is well known that the PLL has an outstanding ability to restrain noise as a result of its narrow-band characteristic. Therefore, it can precisely realize carrier tracking. Nonetheless, in the design procedure of the PLL, the pair of contradictions between lock-in frequency range and tracking precision is always difficult to reconcile. In particular, in deep space

communication, Doppler shift is common and will introduce a considerable frequency offset between transmitter and receiver. In this situation, for the PLL, to increase lock-in frequency range, loop noise bandwidth must be broadened, whereas the precise tracking of its carrier in the condition of a relatively low signal-to-noise ratio (SNR) is dependent on a narrow loop noise bandwidth. So, a large lock-in frequency range (loop noise bandwidth) and a high tracking precision cannot be simultaneously satisfied [2]. In practical designs, a compromise between them is a best choice.

Except for PLL, there are also many kinds of methods referred to as automatic frequency control (AFC) for carrier recovery, such as the frequency recovery loop based on feedback control [3] and the frequency offset estimator (FOE) based on estimation theory [4–6]. They are usually used in burst communication where the speed of carrier recovery must be very quick. However, a fatal flaw for these methods is a relatively low tracking precision. Thus, in low SNR environments, they do not present a perfect performance.

To recover carrier quickly and precisely in the two situations, a large frequency offset and a relatively low SNR, some approaches combining PLL and AFC are proposed to take advantage of their own merits. In [7], a kind of all-digital phase-locked loop (ADPLL) for QPSK combining a first-order frequency recovery loop based on feedback control with a second-order phase-locked loop is proposed. But due to the use of the FD in possession of a sinusoid characteristic (nonlinear characteristic), namely, the second algorithm shown in Table 1, extra noises are introduced into the loop in the circumstance of a low SNR. On the other side, [8] applies fast Fourier transformation (FFT) into the output of the phase discriminator (PD) of QPSK ADCOL, to roughly estimate frequency offset and then speed up the procedure of the carrier recovery of the ADCOL. However, when phase offset is considerable, its PD performs the nonlinear characteristic. On the other hand, for FFT accurate frequency estimation is proportional to the number of its points, which implies that to estimate precisely frequency must consume many logic resources of FPGA. From the above analysis, we can see that PD and FD also influence carrier recovery. Therefore, [9] proposes an all-digital phase-locked loop (ADPLL) taking Hilbert transform and CORDIC algorithm as its PD, resulting in that the locked-in range of the ADPLL is broadened to the sample frequency of the system (Nyquist rate). Given that our design is to combine MLFE with QPSK ADCOL and then use the former to roughly estimate a large frequency offset to assist the latter to lock quickly and precisely, which makes it possible for the ADCOL to operate within the linear range of its PD, we just draw our attention to the linear characteristic of the FD of the MLFE. Thus, a kind of FD which can implement frequency discrimination linearly is proposed. In other words, our proposed design considers not only the merits of the MLFE and the ADCOL, but also the linear characteristic of the FD of the MLFE so that carrier can be recovered quickly and precisely. Furthermore, the lock-in frequency range is also broadened so that the pair of contradictions hardly reconciled for ADCOL can be alleviated. On the other hand, the whole design process of the ADCOL based on FPGA presented by us will also provide a

guideline for the readers anxious to implement an excellent FPGA-based ADCOL for QPSK, which is hardly found in published literatures.

In the following, the design procedure of the ADCOL based on FPGA is presented step by step. Next, our proposed the FD of linear characteristic, MLFE based on phase domain, and the overall design of our QPSK ADCOL are described in Section 3. In Section 4, simulation results and comparative analyses are given. Finally, conclusion and outlook are obtained in Section 5.

2. FPGA-Based All-Digital QPSK Phase-Locked Loop

With respect to QPSK ADCOL, there are three basic components. They are PD, loop filter, and numerical control oscillation (NCO), respectively. Because it is also a kind of ADPLL, the design procedure of it is the same as the normal ADPLL. Therefore, it is indispensable to analyze the design procedure of the normal ADPLL. The normal ADPLL is derived from the result of digitizing analogy PLL. The analysis and design for analogy PLL has been well known. Some monographs have discussed some analogy PLLs which have different orders [10]. As it is sufficient for our QPSK demodulator to use a second-order ADPLL, we just discuss the digitizing procedure of a second-order analogy PLL.

2.1. The Modeling for All-Digital Phase-Locked Loop.

Figure 1(a) shows the phase domain model of a second-order analogy PLL. It consists of the PD modeled by a subtractor with gain k_d , the loop filter which is modeled by a first-order low pass filter, proportion integral filter, with the s -domain transfer function $F(s) = V_o(s)/V_d(s) = (1/s)((\tau_2 s + 1)/\tau_1)$ to minimize the phase noise of the output of the PD, $V_d(s)$, and a voltage control oscillator (VCO) tuned by $V_d(s)$ to make the output phase $\theta_o(s)$ closed to the input phase $\theta_i(s)$, which acts like a radian frequency integrator as a result of $V_d(s) = \Delta\omega t + \Delta\phi$, and have the s -domain transfer function $V(s) = \theta_o(s)/V_c(s) = k_o/s$.

From the above the analysis, a set of equations describing the s -domain transfer functions of the phase domain model of a second-order analogy PLL can be obtained:

$$F(s) = \frac{V_c(s)}{V_d(s)} = \frac{1}{s} \frac{\tau_2 s + 1}{\tau_1}, \quad (1)$$

$$V(s) = \frac{\theta_o(s)}{V_c(s)} = \frac{K_o}{s}, \quad (2)$$

$$H(s) = \frac{\theta_o(s)}{\theta_i(s)} = \frac{K_d F(s) V(s)}{1 + K_d F(s) V(s)} = \frac{2\xi\omega_n s + \omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}, \quad (3)$$

where $\omega_n = \sqrt{k_o k_d / \tau_1}$ is natural radian frequency, $\xi = \tau_2 \omega_n / 2$ is damping factor, k_o is the gain of the VCO, k_d is the gain of the PD, τ_1 and τ_2 are time constants of the proportion integral filter, and $H(s)$ is the s -domain transfer function of the analogy PLL. Please note that the above the equations are just reasonable on the condition that the phase difference

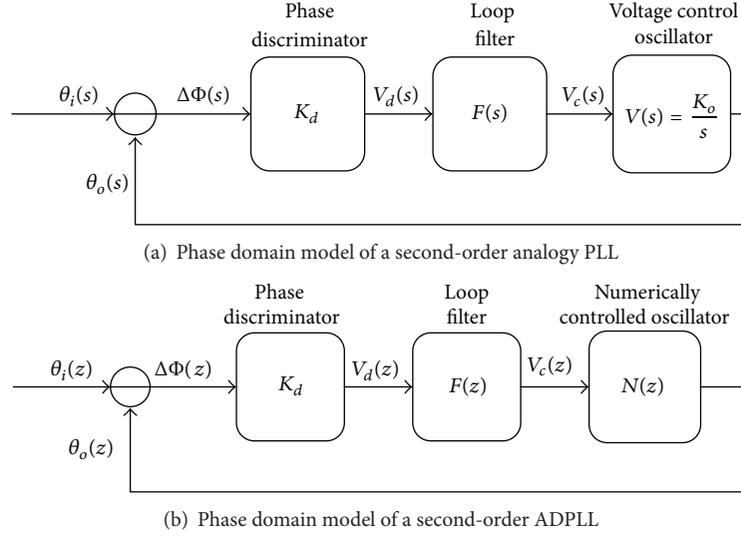


FIGURE 1: The different modes of PLL.

between $\theta_i(s)$ and $\theta_o(s)$ makes it possible for the PD to work within its linear range.

To digitize the analogy PLL, bilinear transformation which is often used to digitize analogy filter [11] is adopted. Let's set the transformation as

$$s = \frac{2}{T_s} \frac{1 - z^{-1}}{1 + z^{-1}}, \quad (4)$$

where T_s is the sample time of a discrete-time system.

Taking (4) into (1) and (2), then we can obtain

$$\begin{aligned} F(z) &= \frac{V_c(z)}{V_d(z)} = C_1 + \frac{C_2}{1 - z^{-1}}, \\ N(z) &= \frac{\theta_o(z)}{V_c(z)} = \frac{k_o z^{-1}}{(1 - z^{-1})}, \end{aligned} \quad (5)$$

where $C_1 = \tau_2/\tau_1 - T_s/2\tau_1$ and $C_2 = T_s/\tau_1$.

Therefore, as shown in Figure 1(b), the model of ADPLL can be acquired. On the basis of Figure 1(b) and the two equations (5); the based-model discrete-time transfer function of the ADPLL can be expressed as

$$\begin{aligned} H(z) &= \frac{\theta_o(z)}{\theta_i(z)} \\ &= \frac{k_d k_o (C_1 + C_2) z^{-1} - k_d k_o C_1 z^{-2}}{1 + [k_d k_o (C_1 + C_2) - 2] z^{-1} + (1 - k_d k_o C_1) z^{-2}}. \end{aligned} \quad (6)$$

2.2. Parameter Calculation of All-Digital Phase-Locked Loop. Form (6), we can see that to obtain the based-model discrete-time transfer function of the ADPLL, the values of the parameters, C_1 , C_2 , k_d , and k_o are needed. However, it is not easy to calculate them after knowing about the model of ADPLL. Nonetheless, none of researches published present the procedure. Thus, in the following we will display how to calculate them.

First of all, the method to acquire C_1 and C_2 is given.

Equation (6) is just the z -domain transfer function based on the model of Figure 1(b) and the two equations, (5). On the other hand, taking (4) into (3), and the z -domain transfer function of the ADPLL based on bilinear transformation can be obtained:

$$\begin{aligned} H(Z) &= \left[4\xi w_n T_s + (w_n T_s)^2 \right] + 2(w_n T_s)^2 z^{-1} \\ &\quad + \left[(w_n T_s)^2 - 4\xi w_n T_s \right] z^{-2} \\ &\quad \times \left[\left[4 + 4\xi w_n T_s + (w_n T_s)^2 \right] + \left[2(w_n T_s)^2 - 8 \right] z^{-1} \right. \\ &\quad \left. + \left[4 + 4\xi w_n T_s + (w_n T_s)^2 \right] z^{-2} \right]^{-1}. \end{aligned} \quad (7)$$

Let us set the denominator of the two z -domain transfer functions of the ADPLL; (6) and (7) obtained by different methods to be equal and the two equations about C_1 and C_2 can be given:

$$\begin{aligned} C_1 &= \frac{1}{K_d K_d} \frac{8\xi w_n T_s}{4 + 4\xi w_n T_s + (w_n T_s)^2} \approx \frac{2\xi w_n T_s}{K_d K_d}, \\ C_2 &= \frac{1}{K_d K_o} \frac{4(w_n T_s)^2}{4 + 4\xi w_n T_s + (w_n T_s)^2} \approx \frac{(w_n T_s)^2}{K_d K_o}, \end{aligned} \quad (8)$$

where the two approximations are just true when $w_n T_s \ll 1$. Namely, assuming that the PD of ADPLL lies in its linear operation range and for ADPLL the characteristic of frequency response is within the range of its passband.

Secondly, on the basis of [12], when $\xi = 0.707$, second-order PLL can meet the optimal value of Wiener theory. So we can get that $\xi = 0.707$.

The third step is to determine w_n .

For PLL, the natural radian frequency w_n determines locked-in frequency range and the performance of suppressing noise. The pair of contradictions between locked-in frequency range (loop noise bandwidth) and tracking precision stem from it.

In the following, we are going to discuss the range of w_n from the two aspects, fast capture bandwidth of ADPLL, and its loop SNR, so as to make a compromise between the locked-in range and the tracking precision.

In the case of ADPLL, there are two kinds of noises, external phase noise and internal phase noise. The external noise caused by additive white Gaussian noise (AWGN) is a main part which has an influence on the performance of the ADPLL, and the internal noise caused by the finite word length effect can be improved by the reasonable selection of word length. Herein, the impact on the selection of w_n is external phase noise. Thus, we just take it into consideration.

The channel of deep-space communication is quite benign, with AWGN being the dominating impairment [1], and thus the phase noise of ADPLL caused by AWGN can be given by [13]

$$\sigma_\theta^2 = \left(\frac{S}{N}\right)_i^{-1} \frac{B_L}{B_i}, \quad (9)$$

where B_i is the bandwidth of input signal of the ADPLL, $(S/N)_i$ is its input SNR, and B_L is loop noise bandwidth.

For the second-order ADPLL taking proportion integral filter as its loop filter, B_L can be expressed as

$$B_L = \frac{w_n}{8\xi} (1 + 4\xi^2). \quad (10)$$

For ADPLL, the ability to suppress noise can be reflected by the loop SNR:

$$\left(\frac{S}{N}\right)_L = \frac{1}{\sigma_\theta^2} = \left(\frac{S}{N}\right)_i \frac{B_i}{B_L}. \quad (11)$$

It determines the size of phase jitter. The result of linear analysis [14] manifests that PLL cannot work normally until $(S/N)_L \geq 6\text{dB}$.

Taking (10) into (11) and considering that ADPLL can operate normally, the upper bound of w_n can be expressed as

$$w_n \leq \left(\frac{S}{N}\right)_i \frac{4\xi B_i}{(1 + 4\xi^2) 3}. \quad (12)$$

The tracking procedure of ADPLL contains frequency tracking and phase tracking, and the former needs longer time than the latter. In design of our ADCRL, however, we firstly use maximum likelihood frequency offset estimator (MLFOE) to assist ADCOL to implement frequency tracking in that for MLFOE the speed of tracking a large frequency offset is superior to ADCOL. Therefore, we just take the phase tracking into consideration in the procedure of designing our ADCOL. In the case of ADPLL, the fast capture bandwidth is defined as the largest frequency offset which ensures that

ADPLL can be locked in the procedure of the phase tracking. It can be expressed as

$$\Delta w_l = 2\xi w_n. \quad (13)$$

To meet the frequency tracking in the presence without the assistance of the MLFOE, the lower bound of w_n is given by

$$w_n \geq \frac{\Delta w_l}{2\xi}. \quad (14)$$

Thus, from (12) and (14), we can get acquire the range of w_n :

$$\frac{\Delta w_l}{2\xi} \leq w_n \leq \left(\frac{S}{N}\right)_i \frac{4\xi B_i}{(1 + 4\xi^2) 3}. \quad (15)$$

Until now, except for K_d and K_o , the parameters needed for calculating C_1 and C_2 have been acquired. Because the K_d and K_o are associated with some practical system parameters, we will discuss about them in the following.

2.3. Parameter Calculation of All-Digital QPSK Phase-Locked Loop Based on FPGA. In the above discussions, we have obtained the corresponding parameters for a normal ADPLL. But, as mentioned at the beginning of Section 2, QPSK ADCOL is also a kind of ADPLL. Thus, the above methods are suitable for QPSK ADCOL.

As shown in Figure 2, QPSK ADCOL is comprised of the PD covered by the shaded area, loop filter surrounded by dashed line, and numerically controlled oscillator (NCO). Now, based on Figure 2, we begin to discuss how to obtain K_d and K_o .

First of all, the analogy-digital converter (ADC) shown in Figure 2 samples the modulated signals from transmitter $RXIN(t)$ and acquires a series of discrete-time signals sampled $RXIN(kT_s)$:

$$\begin{aligned} RXIN(kT_s) &= \sum_{n=0}^k I(nT_s) \cos(w_i nT_s + \theta_i) \\ &\quad + Q(nT_s) \sin(w_i nT_s + \theta_i), \end{aligned} \quad (16)$$

where w_i is the radian frequency of the signals sampled, θ_i is their initial phase, and $I(nT_s)$ and $Q(nT_s)$ are the QPSK signals evaluated as ± 1 in our design.

Secondly, the signals $RXIN(kT_s)$ are mixed with the two outputs of the NCO and then filtered to eliminate the double frequency components generated by the mixing. The two signals filtered can be expressed as

upper branch (in-phase branch):

$$\begin{aligned} I_n(kT_s) &= \frac{1}{2} \sum_{n=0}^k I_n(nT_s) \cos(\Delta w nT_s + \Delta\theta) \\ &\quad - Q_n(nT_s) \sin(\Delta w nT_s + \Delta\theta); \end{aligned} \quad (17)$$

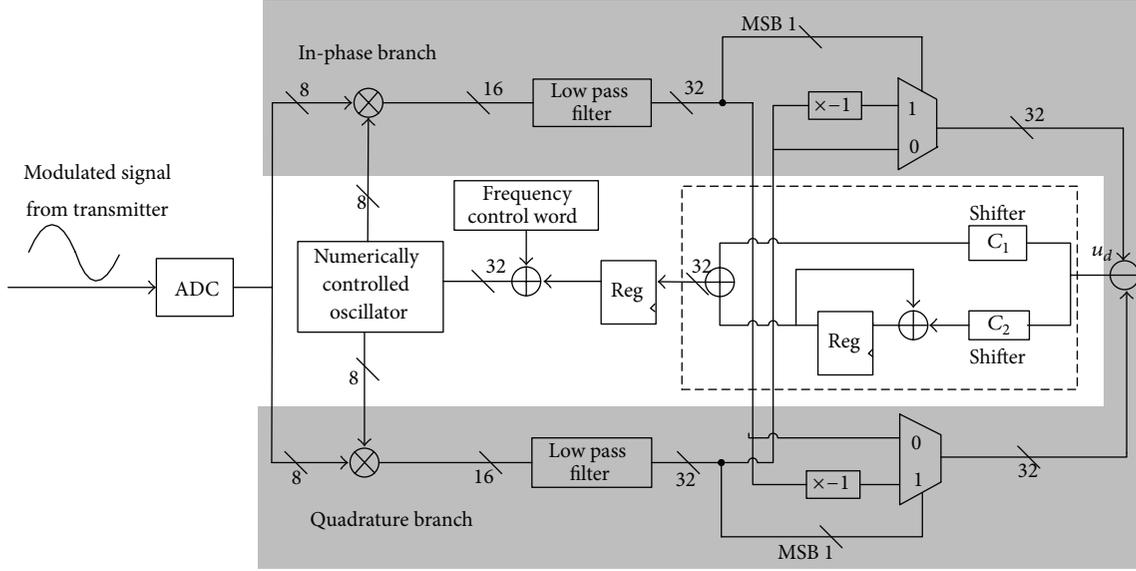


FIGURE 2: Structure diagram of QPSK all-digital phase-locked loop.

bottom branch (quadrature branch):

$$Q_n(kT_s) = \frac{1}{2} \sum_{n=0}^k I_n(nT_s) \cos(\Delta\omega nT_s + \Delta\theta) + Q_n(nT_s) \sin(\Delta\omega nT_s + \Delta\theta), \quad (18)$$

where $\Delta\omega$ is the radian frequency difference between signals sampled and the two outputs of the NCO and $\Delta\theta$ is their initial phase difference.

Finally, the function of the PD of the QPSK ADCOL is implemented by the following:

$$u_d(kT) = \text{sign}(I_n(kT_s)) Q_n(kT_s) - \text{sign}(Q_n(kT_s)) I_n(kT_s). \quad (19)$$

Herein, to save hardware resources of FPGA, the sign decision and the multiplication operation within (19) are replaced by a multiplexer controlled by the most significant bit (MSB) of the outputs of the two low pass filters and an inverter shown in Figure 2 (because the outputs of the two low pass filters are signed numbers). Therefore, the output characteristic of the PD can be obtained:

$$u_d(KT) = \begin{cases} -\sin(\Delta\omega KT_s + \Delta\theta) & -\pi < \Delta\omega KT_s + \Delta\theta < -\frac{3}{4}\pi \\ \cos(\Delta\omega KT_s + \Delta\theta) & -\frac{3}{4}\pi < \Delta\omega KT_s + \Delta\theta < -\frac{\pi}{4} \\ \sin(\Delta\omega KT_s + \Delta\theta) & -\frac{\pi}{4} < \Delta\omega KT_s + \Delta\theta < \frac{\pi}{4} \\ -\cos(\Delta\omega KT_s + \Delta\theta) & \frac{\pi}{4} < \Delta\omega KT_s + \Delta\theta < \frac{3}{4}\pi \\ -\sin(\Delta\omega KT_s + \Delta\theta) & \frac{3}{4}\pi < \Delta\omega KT_s + \Delta\theta < \pi. \end{cases} \quad (20)$$

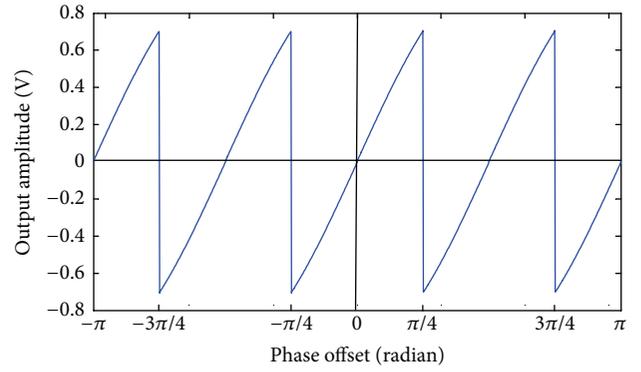


FIGURE 3: Phase offset versus output voltage amplitude of PD.

Based on (20), the output characteristics curve of the PD can be obtained and shown in Figure 3. From Figure 3 we can see that only if the phase offset is within the linear range of the PD ($-\pi/4 \sim \pi/4$), the gain K_d of the PD approximates to 1, namely, the slope of the curve. So this is also one reason why we use MLFE to assist QPSK ADCOL to recover carrier quickly and precisely.

Next, we start with discussing the gain K_o of the NCO.

In [15], the authors implement NCO on a Xilinx FPGA in three types of ways, and the conclusion that the method based on Xilinx ROM is superior to the other two is acquired. Thus, we select the method based on Xilinx ROM to implement our NCO [16].

On the basis of the principle of NCO, the frequency of its output signal can be expressed as

$$f_{\text{out}} = \frac{f_s}{2^N} W_{\Delta\theta}(KT_s), \quad (21)$$

where $f_s = 1/T_s$ is sample frequency, $W_{\Delta\theta(KT_s)}$ is frequency control word, and N is the bit width of the input signal of

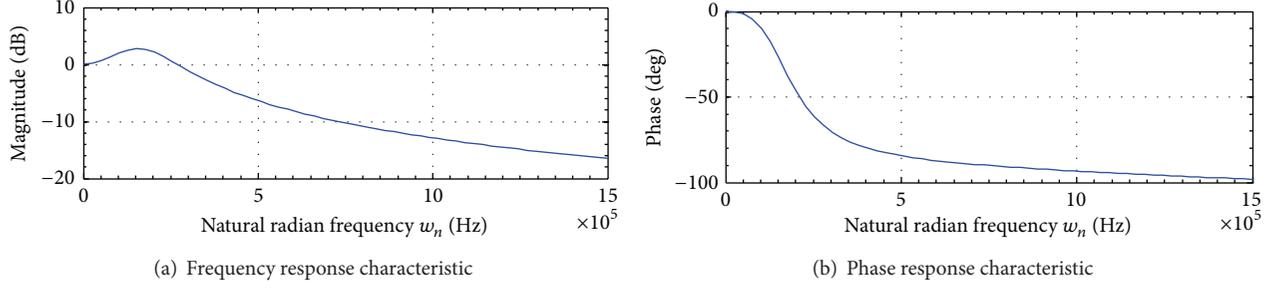


FIGURE 4: QPSK ADCOL characteristic curve.

NCO. In our design, $W_{\Delta\theta(KT_s)} = W_c + W_{ud}$, where W_c is a given value determined by carrier frequency, and its block diagram named as frequency control word can be seen in Figure 2; W_{ud} is the output of the loop filter that is tuned by the frequency offset between transmitter and receiver.

So the radian frequency of NCO is given as

$$W_{\text{out}} = 2\pi f_{\text{out}} = \frac{2\pi f_s}{2^N} W_{\Delta\theta(KT_s)}. \quad (22)$$

On the basis of the model of ADPLL shown in Figure 1(b), NCO is equivalent to a radian frequency integrator. So the output phase of NCO is given as

$$\theta_{\text{out}} = \frac{2\pi f_s T_i}{2^N} W_{\Delta\theta(KT_s)}, \quad (23)$$

where T_i is the update period of the frequency control word $W_{\Delta\theta(KT_s)}$, namely, the sample period of the output of loop filter W_{ud} , and is often set to be $8T_s$.

Therefore, the gain K_o of our NCO is given by

$$K_o = \frac{2\pi f_s}{2^N} 8T_s = \frac{2\pi}{2^{N-3}}. \quad (24)$$

So far, we have obtained all methods to calculate the parameters of the QPSK ADCOL, whereas we must note that, in our QPSK ADCOL based on FPGA shown in Figure 2, an extra gain will be introduced as the result of the changes of the bit width between the inputs and outputs of the different modules. Regardless of the sign bit, the bit width of the inputs of the two multipliers is 7, and the bit width of their outputs is 15. The bit width of the inputs of the two low pass filters is 15, and the bit width of their outputs is 31. Thus the gain of the two multipliers is $2^{15-7} = 2^8$, and the gain of the two low pass filters is $2^{31-15} = 2^{16}$. The rest of the parts showed in Figure 2 have no change in bit width between their inputs and outputs. Therefore, the gain from the changes of bit width is $2^{8+16} = 2^{24}$, and (8) should be rectified as

$$\begin{aligned} C_1 &= \frac{1}{k_d k_o 2^{24}} \frac{8\xi w_n T_s}{4 + 4\xi w_n T_s + (w_n T_s)^2} \approx \frac{2\xi w_n T_s}{k_d k_o 2^{24}}, \\ C_1 &= \frac{1}{k_d k_o 2^{24}} \frac{4(w_n T_s)^2}{4 + 4\xi w_n T_s + (w_n T_s)^2} \approx \frac{(w_n T_s)^2}{k_d k_o 2^{24}}. \end{aligned} \quad (25)$$

On the basis of the core idea of software defined radio, the parts of digital signal processing should be closed to the front end of radio frequency (RF) as much as possible. Therefore, we make our QPSK ADCOL operate in intermediate frequency (IF), namely, the ADC sample frequency ($f_s = 26$ MHz) and carrier frequency (the output frequency of NCO) $f_c = 4$ MHz.

On the other hand, To ensure that our QPSK ADCOL can normally operate under the conditions of a large frequency offset and a low SNR, let us set the fast capture bandwidth $\Delta\omega_l$ to be 100 KHZ the least, and the input SNR (S/N) to be 1 dB. We have known that the gain of PD is $K_d = 1$, and the damping factor is $\xi \approx 0.707$. To decrease internal phase noise caused by word length effect, we set the bit width of input signal of NCO to be $N = 32$. Therefore, based on (15), (24), and (25), the range of the natural radian frequency w_n can be obtained and the one of the values is chosen as

$$w_n = 2\pi \times 150 \times 10^3 = 0.942 \times 10^6 \text{ (rad/s)}. \quad (26)$$

Next, take $K_d K_o 2^{24}$, (25), and (26) into (6), we can obtain the model-based discrete-time transfer function of our QPSK ADCOL:

$$H(z) = \frac{0.053z^{-1} - 0.051z^{-2}}{1 - 1.947z^{-1} + 0.949z^{-2}}. \quad (27)$$

From (27), the poles of our QPSK ADCOL can be obtained. They are $0.973 \pm 0.036i$. Based on the theory of the stability of discrete system, the system is stable if all poles are located inside the unit circle. Therefore, our QPSK ADCOL is stable.

Now, the frequency response characteristic and phase response characteristic of our QPSK ADCOL can be acquired and shown in Figures 4(a) and 4(b), respectively.

From Figures 4(a) and 4(b), we can see that when sample frequency is 26 MHz, the passband of our QPSK ADCOL ranges from 0 to 200 KHZ (namely, $w_n T_s \ll 1$ is true) and its margin of phase is 120 degree below. Thus, this also indicates that our QPSK ADCOL meets the conditions of the stability of negative feedback control system. What is more, Figure 4(a) also displays that QPSK ADCOL is of the property of low pass.

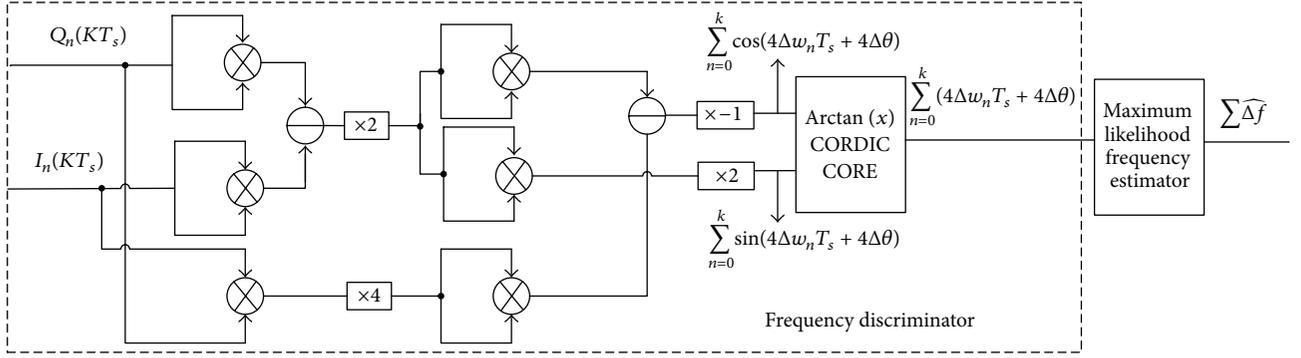


FIGURE 5: The block diagram of the FD and MLFE (MLFOE).

TABLE 1: Two kinds of typical frequency discriminators.

The algorithm used	Frequency offset output	Output characteristic	Hardware complexity
$\frac{\text{sig}(\cdot) \times \text{cross}}{(k-1)T_s - kT_s}$	$\frac{\sin[2(\theta_{k-1} - \theta_k)]}{(k-1)T_s - kT_s}$	Nonlinear	Moderate
$\frac{\text{cross}}{(k-1)T_s - kT_s}$	$\frac{\sin(\theta_{k-1} - \theta_k)}{(k-1)T_s - kT_s}$	Nonlinear	Simple

$$* \text{dot} = I_n(KT_s) \times I_n((K-1)T_s) - Q_n(KT_s) \times Q_n((K-1)T_s).$$

$$* \text{cross} = I_n(KT_s) \times Q_n((K-1)T_s) - I_n((K-1)T_s) \times Q_n(KT_s).$$

3. The Design of Phase Domain Maximum Likelihood Frequency Estimator and Its Frequency Discriminator

3.1. The Frequency Discriminator of the Linear Characteristic. To use MLFE to obtain an accurate frequency offset, the performance of its FD is a key factor which must be considered. Two kinds of FDs used widely are summarized in Table 1 [17]. From Table 1, we can see that they are all of nonlinear characteristic. Because in the case of $\sin(\Delta\theta)$, $\sin(\Delta\theta) \approx \Delta\theta$ is true only if $\Delta\theta$ varies within a small range. However, in deep space communication (or in the condition that a large Doppler shift is common), the approximation is hardly possible. On the other hand, to discriminate phase offset, the dot product and cross-product from two sample points separated by a sample interval must be conducted, and their results divide by the sample interval ($\theta = \omega t = 2\pi f t$, namely, $f = \theta/2\pi t$).

Therefore, we introduce a kind of FD in possession of linear characteristic and the ability to acquire the corresponding signals of our ADCOL so as to transform them into the inputs of the MLFE. Its block diagram surrounded by the dashed line is shown in Figure 5.

From Figure 5, we can see that after a series of transformations for the two equations, (17) and (18), the two signals of the front end of the CORDIC algorithm block can be obtained. They are

$$\begin{aligned} \cos(4\Delta\omega KT_s + 4\Delta\theta) &= \cos(4(2\pi\Delta f KT_s + \Delta\theta)), \\ \sin(4\Delta\omega KT_s + 4\Delta\theta) &= \sin(4(2\pi\Delta f KT_s + \Delta\theta)). \end{aligned} \quad (28)$$

Feed them into the CORDIC algorithm block which implements the algorithm $\tan^{-1}(\sin(4\Delta\omega KT_s + 4\Delta\theta)/\cos(4\Delta\omega KT_s + 4\Delta\theta))$, and the phase offset $4\Delta\omega KT_s + 4\Delta\theta = 4(2\pi\Delta f KT_s + \Delta\theta)$ can be acquired. After that, the MLFE will be used to estimate the frequency offset Δf . It is clear that this is a procedure of resolving linearly frequency offset. Because the CORDIC algorithm can easily be implemented on FPGA just using sniffers and add operations [18], we just discuss about how to implement MLFE on FPGA using as few logic resources as possible.

3.2. Phase Domain Maximum Likelihood Frequency Estimation Algorithm. In the case of the FD shown in Figure 5, a mapping transformation from Cartesian domain to phase domain can be realized by CORDIC core and expressed as

$$\tilde{x} = \begin{cases} \arctan\left(\frac{Q_k}{I_k}\right) & I_k > 0 \\ \arctan\left(\frac{Q_k}{I_k}\right) - \pi & Q_k < 0, I_k \leq 0 \\ \arctan\left(\frac{Q_k}{I_k}\right) + \pi & Q_k > 0, I_k \leq 0, \end{cases} \quad (29)$$

where $I_k = \cos(4\Delta\omega KT_s + 4\Delta\theta)$ and $Q_k = \sin(4\Delta\omega KT_s + 4\Delta\theta)$. \tilde{x}_k is the discrete phase of k th sample point. The amendment of $\pm\pi$ is due to that the output range of our CORDIC core is within $(-\pi \sim \pi)$.

To use MLFE for estimating frequency offset, we first need to obtain the discrete phases of M continuous sample points (\tilde{x}_k , $0 \leq k \leq M-1$) and then take the first sample point \tilde{x}_0

as initial reference point to obtain M absolute phases, which can be expressed as

$$x_k = x_{k-1} + \begin{cases} \tilde{x}_k - \tilde{x}_{k-1} & |\tilde{x}_k - \tilde{x}_{k-1}| < \pi \\ \tilde{x}_k - \tilde{x}_{k-1} + 2\pi & \tilde{x}_k - \tilde{x}_{k-1} < -\pi \\ \tilde{x}_k - \tilde{x}_{k-1} - 2\pi & \tilde{x}_k - \tilde{x}_{k-1} > \pi \end{cases} \quad (30)$$

$$1 \leq k < M, \quad x_0 = \tilde{x}_0,$$

where \tilde{x}_k (the output of our CORDIC core) is the discrete phase of k th sample point, and it ranges from $-\pi$ to π . x_k is an absolute phase of k th sample point, which takes \tilde{x}_0 as the point of reference, and has no the limitation of phase ranging from $-\pi$ to π . The amendment of $\pm 2\pi$ is due to the phase difference between the $(k-1)$ th sample point, \tilde{x}_{k-1} , and the k th, \tilde{x}_k , crosses over a cycle (2π) of the output of our CORDIC core.

After that, a recursive formula of the M absolute phases can be obtained:

$$x_k = 4 \times (2\pi K T_s \Delta f + \Delta\theta + n_k) \quad 0 \leq K \leq M-1, \quad (31)$$

where n_k is the phase noise caused by AWGN (for the sake of simplicity, we neglect it in (16), (17), (18), and (28)), T_s is the sample frequency, Δf is the frequency offset between the output of the NCO and the modulated signal from transmitter, and $\Delta\theta$ is initial phase difference of the two signals. When SNR is as low as 10 dB, numerical results have been demonstrated that n_k can be considered as the Gaussian approximation accurate which has a zero mean and variance σ^2 [19].

Let's set $2\pi K T_s \Delta f + \Delta\theta + n_k$ to be z_k , namely,

$$z_k = 2\pi K T_s \Delta f + \Delta\theta + n_k \quad 0 \leq k \leq M-1. \quad (32)$$

Equation (35) can be also written in vector form:

$$Z = \Delta f 2\pi T_s \alpha + \Delta\theta \beta + V, \quad (33)$$

where

$$Z = \begin{bmatrix} z_0 \\ z_1 \\ \vdots \\ z_{M-2} \\ z_{M-1} \end{bmatrix} = \Delta f 2\pi T_s \begin{bmatrix} 0 \\ 1 \\ \vdots \\ M-2 \\ M-1 \end{bmatrix} + \Delta\theta \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} n_0 \\ n_1 \\ \vdots \\ n_{M-2} \\ n_{M-1} \end{bmatrix}. \quad (34)$$

Consequently, Z is a Gaussian random vector with probability density function:

$$f_z(Z) = \frac{1}{\sqrt{(2\pi)^M \sigma^M}} \exp \left[-\frac{\|Z - \Delta f 2\pi T_s \alpha - \Delta\theta \beta\|^2}{2\sigma^2} \right], \quad (35)$$

where $\|\cdot\|^2 = (Z - \Delta f 2\pi T_s \alpha - \Delta\theta \beta)^T (Z - \Delta f 2\pi T_s \alpha - \Delta\theta \beta)$.

The maximum likelihood estimators Δf and $\Delta\theta$ can be obtained by equating the gradient $\nabla_{\Delta f, \Delta\theta} \log f_z(z)$ to zero and solving a two-dimensional linear system:

$$\Delta \widehat{f}(z) = \frac{12}{2\pi T_s (M-1) M (M+1)} F^T Z, \quad (36)$$

$$\widehat{\theta}(z) = \frac{6}{M(M+1)} \Theta^T Z, \quad (37)$$

where

$$Z = \begin{bmatrix} z_0 \\ z_1 \\ \vdots \\ z_{M-2} \\ z_{M-1} \end{bmatrix}, \quad F = \begin{bmatrix} -\frac{M-1}{2} \\ -\frac{M-2}{2} + 1 \\ \vdots \\ \frac{M-1}{2} - 1 \\ \frac{M-1}{2} \end{bmatrix}, \quad (38)$$

$$\Theta = \begin{bmatrix} \frac{2M-1}{3} \\ \frac{2M-1}{3} - 1 \\ \vdots \\ \frac{2M-1}{3} - (M-2) \\ \frac{2M-1}{3} - (M-1) \end{bmatrix}.$$

The maximum likelihood estimators in (36) and (37) are minimum variance unbiased estimations achieving the Cramer Rao Bound [19, 20]. On the other side, the higher the estimation accuracy is, the larger the sample points M and SNR are.

Please note that (31) is $4Z_k$. If we use (36) to estimate frequency offset Δf , the value of estimation must multiply by 4. In the case of FPGA, the multiplication of 2^n just needs to shift n bits towards the left.

On the other hand, in part 3, Section 2, we select update period of frequency control word of the NCO to be $8T_s$. Therefore, to enable frequency offset estimation block shown in Figure 5 and the QPSK ADCOL shown in Figure 2 to operate as synchronously as possible, we set the number of the sample point of MLFE to be $M = 8$. Therefore, based on (36), (34), (32), and (31), we can obtain the frequency offset estimated:

$$\widehat{\Delta f} = 4\Delta \widehat{f}(z) = 4 \frac{12}{2\pi T_s (8-1) 8 (8+1)} \times \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_6 \\ x_7 \end{bmatrix}, \quad (39)$$

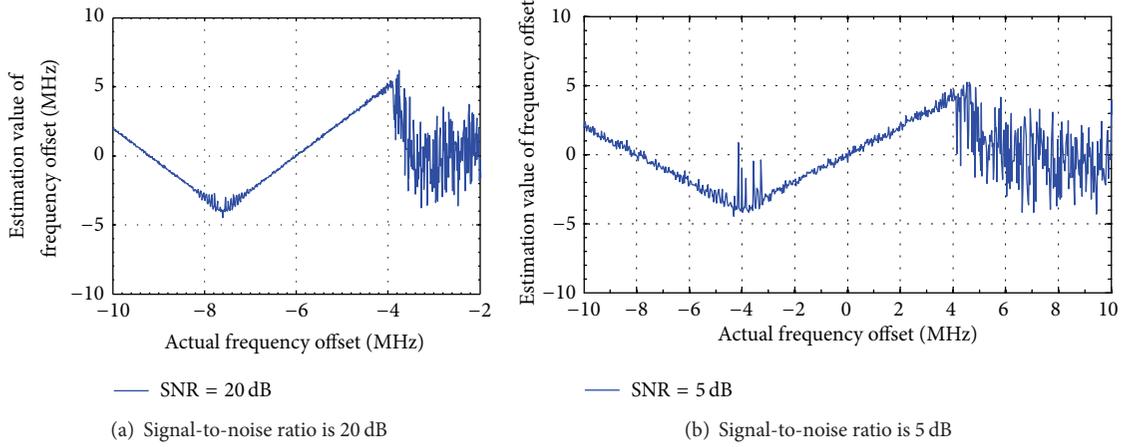


FIGURE 6: Actual frequency offset versus estimation frequency offset under different signal-to-noise ratios.

where $T_s = 1/26$ MHz is sample frequency and $x_k, 0 \leq k \leq 7$ is absolute phase generated by (30).

Using $\widehat{\Delta f}$ to assist the QPSK ADCOL shown in Figure 2 to recover carrier quickly, $\widehat{\Delta f}$ must be transformed into frequency control world of the NCO. On the basis of (21), we can obtain frequency control world of $\widehat{\Delta f}$ that is

$$w_{\Delta f} = \frac{\widehat{\Delta f} 2^N}{f_s}. \quad (40)$$

Taking (39) into (40), we can get

$$w_{\Delta f} = \frac{2^N}{21\pi} \left[-\frac{8-1}{2}, -\frac{8-1}{2} + 1, \dots, \frac{8-1}{2} - 1, \frac{8-1}{2} \right] \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_6 \\ x_7 \end{bmatrix}, \quad (41)$$

where $x_k, 0 \leq k \leq 7$, are signed decimals, which are expressed as the fixed-point number with 3 bits' integer number and 29 bits' decimal. $N = 32$ is bit width of input signal of the NCO. Because frequency control world of NCO is an integer number, the result of (41) should multiply by 2^{-29} so as to eliminate the affection of the decimal expressed by 29 bits' binary format. Therefore, actual frequency control world should be

$$w_{\Delta f} = 0.1213 \left[-\frac{8-1}{2}, -\frac{8-1}{2} + 1, \dots, \frac{8-1}{2} - 1, \frac{8-1}{2} \right] \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_6 \\ x_7 \end{bmatrix}. \quad (42)$$

Due to $0.1213 \approx 2^{-3} - 2^{-8}$, maximum likelihood frequency estimation can be implemented just using shifters and multiply-accumulate units.

In deep-space communication, to confirm that the signals buried by noise can be successfully detected by receiver, a parameter named link margin is used to specify minimal SNR of the received signals. In the practical design of the communication systems, its value usually ranges from 3 dB to 6 dB [21]. In our design, we select two types of link margins to investigate the performance of our frequency offset estimator. They are the low link margin of 5 dB and the high link margin of 20 dB, respectively. Therefore, we simulate our MLFE under the condition of sample frequency $f_T = 26$ MHz and carrier frequency $f_c = 4$ MHz by MATLAB. The simulation results are shown in Figure 6.

From Figure 6, we can see that although the estimation rang of the frequency offset decreases along with the decline of SNR, the range still approximates -2 MHz \sim 2 MHz under SRN = 5 dB. In the case of low and medium earth orbiting satellite where the greatest Doppler shifts are ± 100 KHZ and ± 200 KHZ [22], respectively, the rang completely meets as well.

3.3. Entire Design of All-Digital Carrier Recovery Loop of QPSK. Figure 7 shows the block diagram of our ADCRL. It consists of the MLFOE in shadow and QPSK ADCOL surrounded by dashed line, respectively.

First of all, the frequency offset will be estimated roughly and then transformed into the frequency control words of the NCO by MLFE to speed up QPSK ADCOL to quickly implement the tracking of the carrier.

Secondly, with the assistance of MLFOE, the QPSK ADCOL is locked quickly and then starts with tracking the carrier precisely.

From Figure 7, we can see that the bit width of the outputs of the two low pass filters is 32. If we apply directly the width to MLFOE, the cost of hardware resource for FPGA is considerable. So in our design, a truncated bit width will be used. To ensure that the impact of the truncation on the performance of our system is minimal, a simulation is conducted, which uses different bit widths for the two input signals of MLFOE (the bit widths of the outputs of the two low pass filters) and the widths range from 8 to 32 bits to test

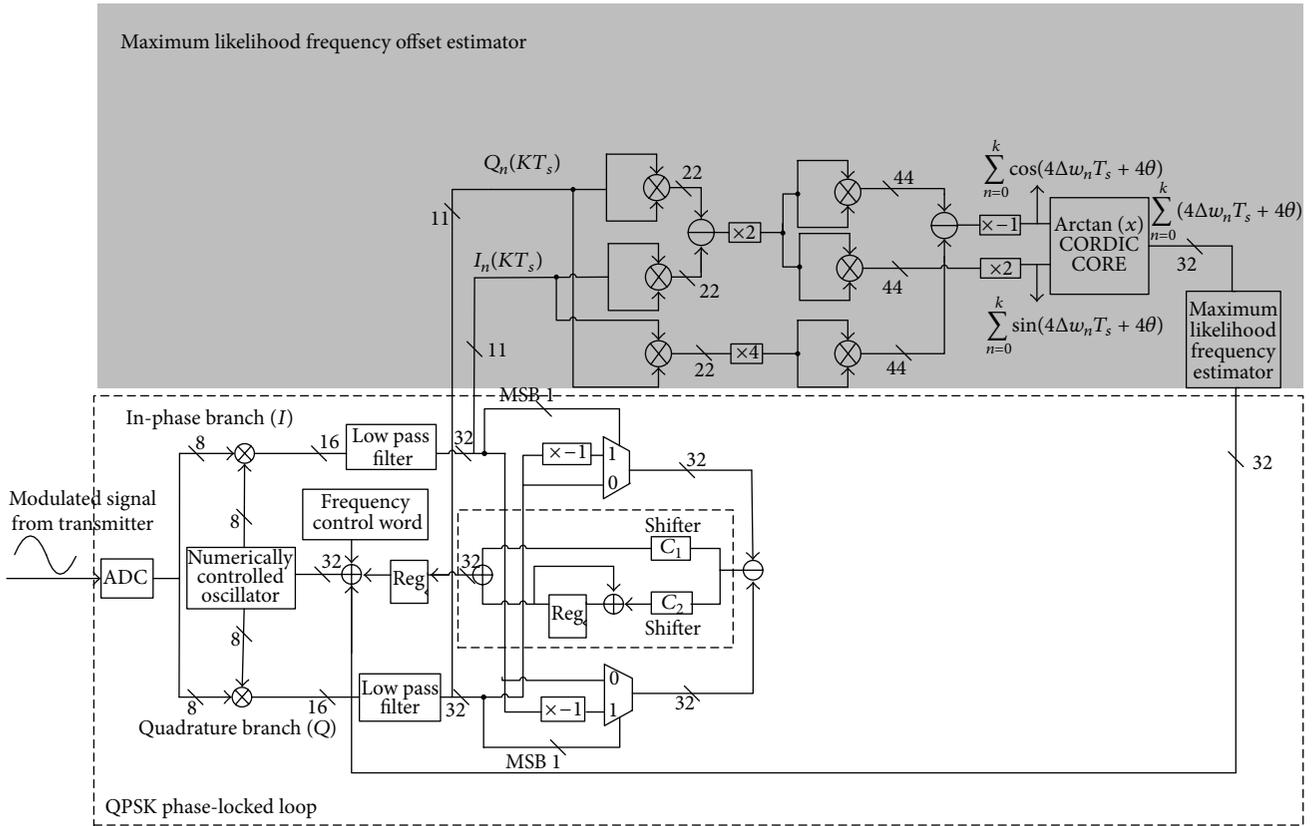


FIGURE 7: Block diagram of QPSK ADCRL.

which one is the best for the frequency offset error $\Delta f = 100$ KHZ under the situations of SNR = 20, 10, 5 dB. The simulation result is shown in Figure 8. From Figure 8, it is clear that when the bit width is equal to 11, the frequency estimation value of the MLFOE is almost the same as that of having bit width equal to 32 under the three types of SNRs. Thus, we set the bit width of the two inputs of MLFOE to be 11.

4. Simulation Results and Comparative Analysis

To verify that the proposed architecture is valid, the FPGA simulation tool, ModelSim SE 6.5, is used to observe the implementation performance of our proposed architecture based on FPGA. On the other hand, MATLAB is also used to generate the QPSK modulated signals with a frequency offset under the conditions of different SNRs and process the simulation results generated by the ModelSim so as to have a better visual comparison, resulting from that the outputs of the ModelSim that are just some values of decimal or binary format.

The data streams of the entire simulation procedure are shown in Figure 9.

The input stream generated, quantized and stored into the textfile, Textfile_A, by MATLAB, is QPSK modulated signals with the following specifications:

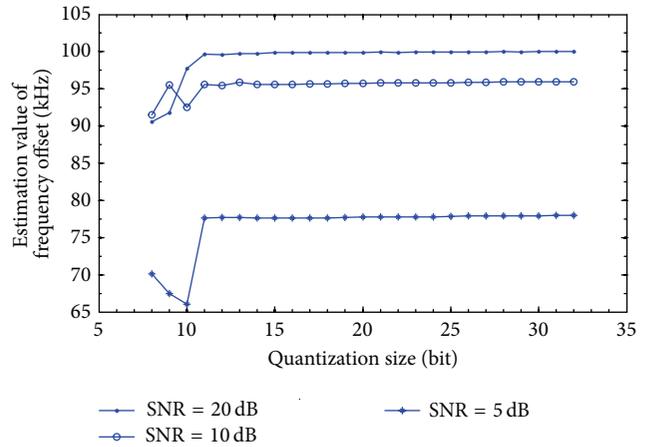


FIGURE 8: The quantization level of the two input signals of MLFOE under the condition of the frequency offset error $\Delta f = 100$ KHZ and SNR = 20, 10, 5 dB.

- (i) the number of symbols randomly evaluated as ± 1 is equal to 1000;
- (ii) the frequency of symbols $f_b = 80$ KHZ;
- (iii) the frequency of carrier $f_c = 3.9$ MHZ, namely, frequency offset $\Delta f = 100$ KHZ;
- (iv) SNRs are 20 dB and 5 dB;

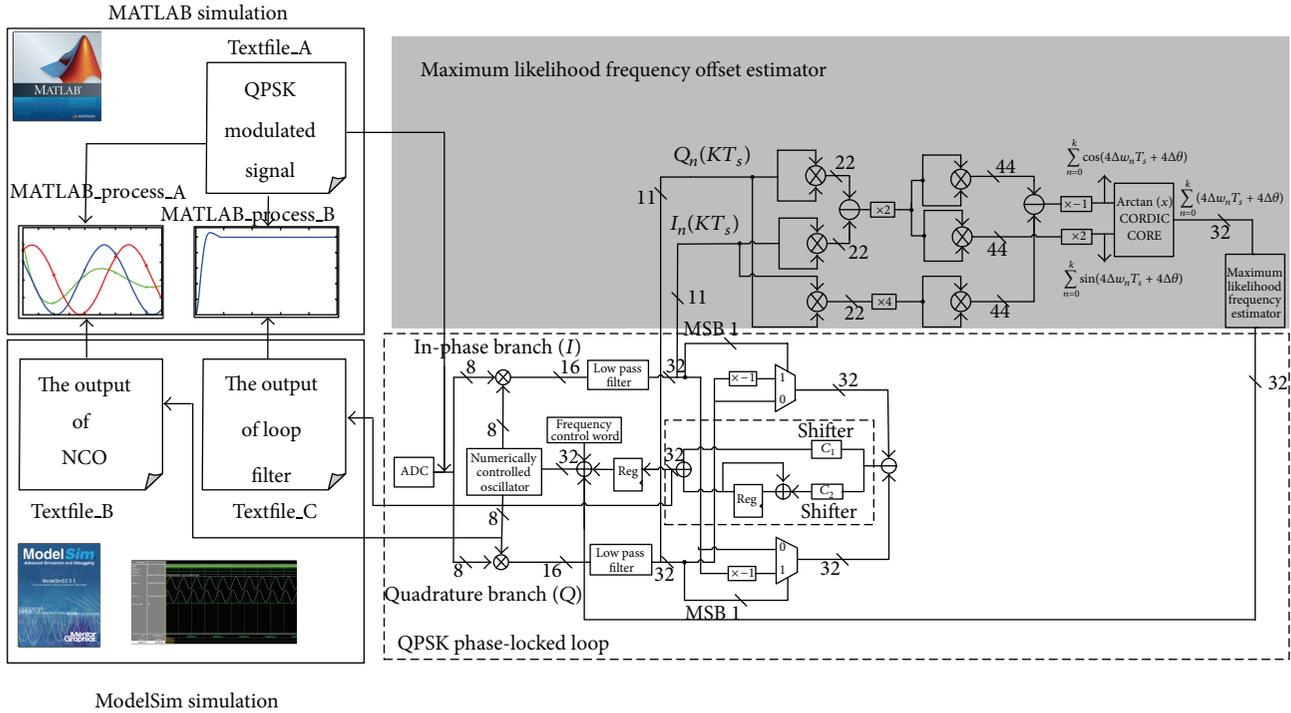


FIGURE 9: The data stream of entire simulation procedure.

- (v) quantization level of the 1000 QPSK modulated signals is 8 bits to imitate the input signals of 8-bit ADC;
- (vi) sample frequency $f_s = 26$ MHz.

The output streams generated and stored into another two textfiles, Textfile_B, and Textfile_C by ModelSim are the outputs of the NCO and the loop filter with the following specifications:

Textfile_B:

- (i) the frequency of carrier $f_c = 4$ MHz, namely output frequency of NCO;
- (ii) sample frequency $f_s = 26$ MHz;
- (iii) the quantization level of the output amplitude of the NCO evaluated as ± 1 is 8 bits;

Textfile_C:

- (i) sample frequency $f_s = 26$ MHz;
- (ii) the quantization level of the output of the loop filter is 32 bits.

Textfile_A is generated by MATLAB before starting ModelSim simulation and then fed into QPSK ADCRL (to imitate the output of the ADC) in the procedure of the simulation of ModelSim, when Textfile_B and Textfile_C are simultaneously generated by ModelSim. At the end of the simulation of the ModelSim, the two files from the ModelSim, Textfile_B and Textfile_C, will be read into MATLAB in order to further process in the two procedures, MATLAB_process_A and MATLAB_process_B.

Because the traditional measure of the performance of PLL is based on locked-in time, steady-state phase error, and

locked-in frequency range, our QPSK ADCRL also uses the methods.

Based on the principle of our QPSK ADCRL displayed in Figure 7, when there exists the frequency offset between the input of the QPSK modulated signals (Textfile_A) and the local carrier signals (the output of NCO), the offset can be obtained through taking the output of the loop filter (Textfile_C) into (21). In our design, the carrier frequency of the input modulated signals is 3.9 MHz and the frequency of the output of the NCO is 4 MHz. Therefore the frequency offset is 100 KHZ.

If our QPSK ADCRL is locked, the frequency offset evaluated through taking the output of loop filter into (21) will approximate to 100 KHZ. At the same time, the output phase of the NCO (Textfile_B) should be equal or approximate to the phase of the QPSK modulated signals (Textfile_A).

On the basis of above discussion, as shown in Figures 10, 11, 12, and 13, (a) and (b) are the outputs of the loop filter from ModelSim simulation, and its results processed by MATLAB (MATLAB_process_B shown in Figure 8), respectively. (c) is the result of the phase comparison of the two signals, QPSK modulated signals, and the outputs of the NCO, which is from the simulation of ModelSim and then processed by MATLAB (MATLAB_process_A shown in Figure 9).

Figures 10 and 11 are the simulation results of the classic QPSK ADCOL shown in Figure 2 without the assistance of MLFOE shown in Figure 6 under the two conditions of SNR = 20 and 5 dB.

From the two figures, Figures 10(b) and 10(c), we can see that when SNR is equal to 20 dB, the tracking time is about 0.15 ms and the maximal steady-state phase error approximates to 0 degree.

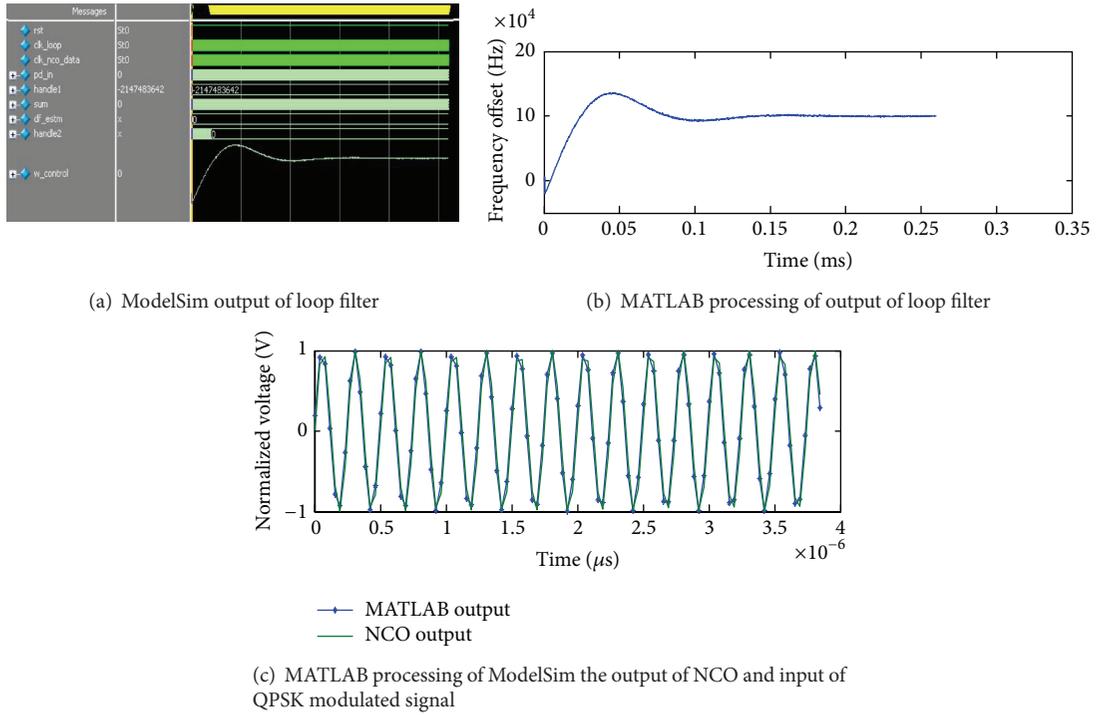


FIGURE 10: The joint simulation result of ModelSim and MATLAB under the condition of SNR = 20 dB for the QPSK ADCOL.

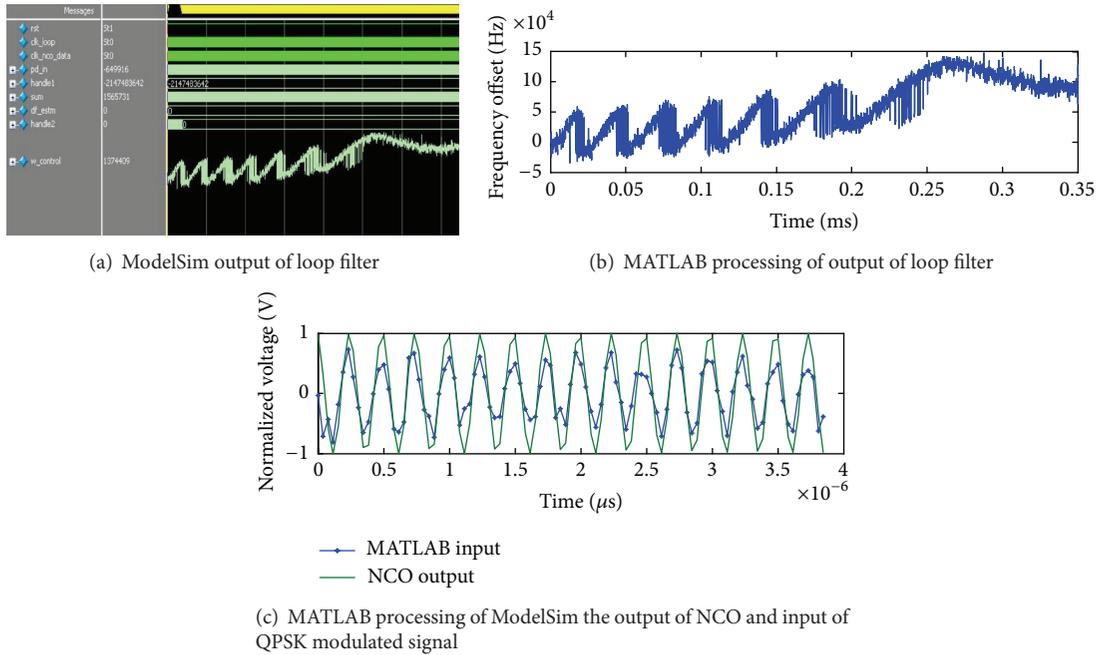


FIGURE 11: The joint simulation result of ModelSim and MATLAB under the condition of SNR = 5 dB for the QPSK ADCOL.

As shown in the two figures, Figures 11(b) and 11(c), when SNR is equal to 5 dB, the tracking time is about 0.3 ms and the maximal steady-state phase error approximates to 2 degree.

Figures 12 and 13 are the simulation results under the two conditions of SNR = 20 and 5 dB, where the MLFOE has been enabled.

In contrast to Figures 10 and 11, after enabling the MLFOE, from the two figures, Figures 12(b) and 12(c), we can see that when SNR is equal to 20 dB, the locked-in time is about 0.05 ms and the maximal steady-state phase error also approximates to 0 degree. As shown in the two figures, Figures 13(b) and 13(c), when SNR is equal to 5 dB, the lock-in

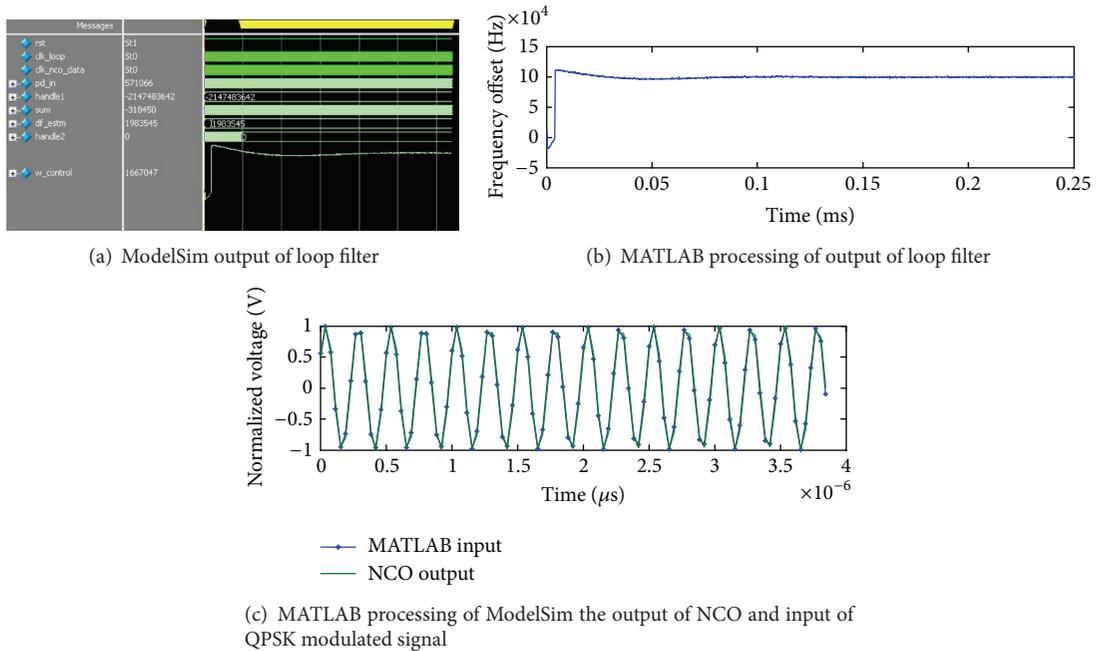


FIGURE 12: The joint simulation result of ModelSim and MATLAB under the condition of SNR = 20 dB for our QPSK ADCRL.

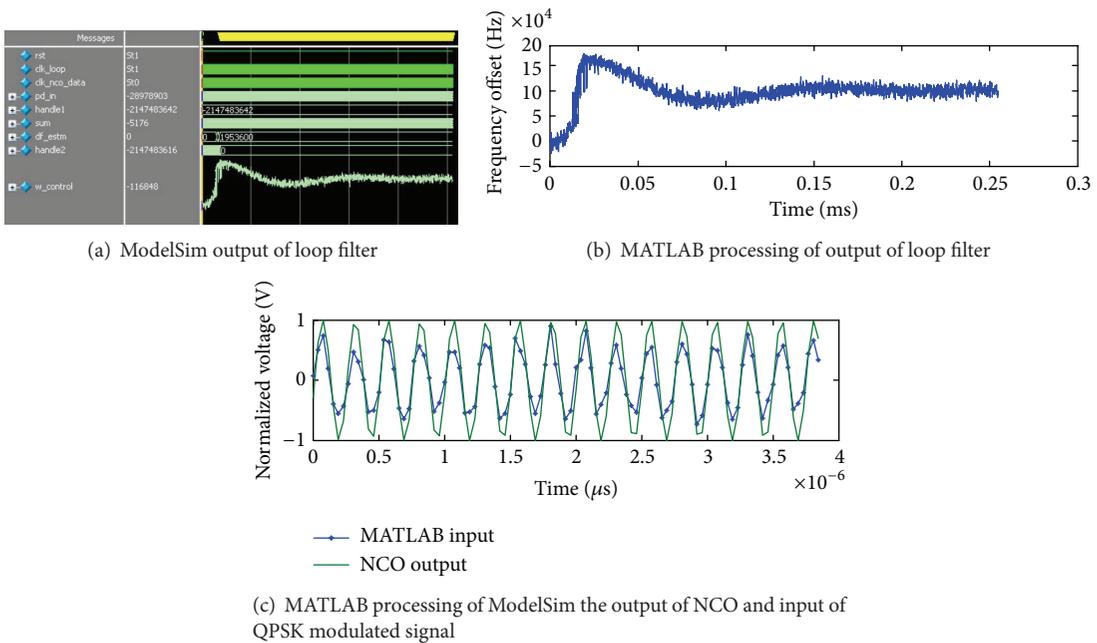


FIGURE 13: The joint simulation result of ModelSim and MATLAB under the condition of SNR = 5 dB for our QPSK ADCRL.

time is about 0.1 ms and the maximal steady-state phase error approximates to 3 degree.

On the other hand, in the case of the four figures, Figures 10(b), 11(b), 12(b), and 13(b), when QPSK ADCRL is locked all frequency offset evaluated by the output of loop filter are equal to 100 KHZ.

From the above simulation results, it is clear that after the MLFOE is enabled, whether SNR is high or not; the performance of our QPSK ADCRL in locked-in time is two times

faster than that of the QPSK ADCOL without the assistance of the MLFOE, while the maximal steady-state phase error is almost stable.

On the other hand, in the two conditions of SNR = 20 dB and 5 dB, we test the maximal frequency offset which enables our QPSK ADCRL and the QPSK ADCOL without the assistance of MLFOE to be locked and the corresponding locked-in time and steady-phase error. The results are shown in Table 2. From Table 2, we can see that after the MLFOE

TABLE 2: The performance advantage of our QPSK ADCRL in locked-in frequency range.

SNR	Architecture	Maximal locked-in Frequency Range (KHZ)	Locked-in time (ms)	Steady-phase error (degree)
20 db	QPSK ADCOL	± 170	0.18	Approximating to 0
20 db	Our QPSK ADCRL	± 680	0.05	Approximating to 0
5 db	QPSK ADCOL	± 120	0.4	Approximating to 2
5 db	Our QPSK ADCRL	± 510	0.13	Approximating to 4

TABLE 3: The hardware cost of the different modules for our QPSK ADCRL.

Module	Number of slice registers	Number of slice LUTs	Number used as logic
QPSK ADCOL	174 out of 28800	580 out of 28800	612 out of 28800
MLFOE	FD	875 out of 28800	790 out of 28800
	MLFE	603 out of 28800	592 out of 28800
Total hardware cost	1652 out of 28800	1988 out of 28800	1994 out of 28800

TABLE 4: The power consumption of our QPSK ADCRL for different operating frequencies.

Clock frequency (MHZ)	Dynamic power (W)	Quiescent power (W)	Total power (W)	Junction temp. (C)
312	0.060	0.526	0.594	50.9
250	0.058	0.526	0.583	51
200	0.048	0.525	0.574	51
150	0.039	0.525	0.564	50.9

is enabled, except for the advantage in locked-in time, the locked-in frequency range of our QPSK ADCRL is four times wider than that of the QPSK ADCOL. Therefore, our QPSK ADCRL can alleviate the pair of contradictions between lock-in frequency range and tracking precision, which are hardly reconciled for the QPSK ADCOL.

There is no doubt that our QPSK ADCRL has a more robust performance than the QPSK ADCOL without the MLFOE. What is more, to obtain the same improvement in performance as our ADCRL for the existing QPSK ADCOL [23, 24], the simple revision is to add the MLFOE shown in Figure 4 to the QPSK ADCOLs.

Finally, in order to acquire their hardware cost for the different modules of our QPSK ADCRL based on FPGA, the FPGA synthesis tool, ISE Design Suite 12.2, from FPGA vendor Xilinx is used, and its chip of Virtex5 family, XC5VLX50, which supports dynamic reconfiguration technology seen as a core technology to implement SDR [25] on FPGA is selected. The synthesis results are given in Table 3. From Table 3, we can see that although the MLFOE consumes more logic resources than QPSK ADCOL (because many multiplication operations are used, and a pipelined CORDIC architecture [26] is adopted so as to meet the requirement of the latency time), this is worthy for some applications which need a more robust QPSK ADCRL.

On the other hand, we can also see that the hardware cost of the whole QPSK ADCRL just make up a small part of logic resources for the FPGA chip selected.

Except for the hardware cost of our QPSK ADCRL, we also investigate its power consumption for the different

operating frequencies by the power analysis tool from Xilinx, XPower, which is also important very much in some applications where communication systems need to work continuously by means of a portable power source. The results are shown in Table 4. From Table 4, we can see that when our QPSK ADCRL operates under the condition of maximum clock frequency, 312 MHZ, which is from the logic synthesis's result of ISE 12.2, the total power is just 0.594 (W).

5. Conclusion and Outlook

In this paper, an efficient QPSK ADCRL is proposed, and a systematic procedure of designing the carrier recovery loop based on FPGA is displayed. On the other hand, a FD in possession of linear characteristic is introduced to supply a more precise frequency offset to MLFE, which is implemented just using shifters and multiply-accumulate units to estimate the frequency offset and assist QPSK ADCOL to lock quickly. The joint simulation results of ModelSim and MATLAB has proved that our proposed architecture can smoothly operate on FPGA and its performances in locked-in time and locked-in range are more excellent than the classic QPSK ADCOL. Synthesis result has shown that the hardware cost of FPGA for our QPSK ADCRL is very few, and the result of the power analysis also has proved that our design is valid in power consumption.

Looking at the future, the exploration aiming at deep space must be the tendency of the human development, and the FPGA-based soft defined radio suitable for the

environment will be also more widely applied and further studied.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] J. Yuen, *Autonomous Software-Defined Radio Receivers for Deep Space Applications*, Wiley, 2006.
- [2] N. Kim and I. Ha, "Design of ADPLL for both large lock-in range and good tracking performance," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 46, no. 9, pp. 1192–1204, 1999.
- [3] C. Cahn, "Improving frequency acquisition of a Costas loop," *IEEE Transactions on Communications*, vol. 25, no. 12, pp. 1453–1459, 1977.
- [4] E. Frantzeskakis, C. Papathanasiou, D. Doumenis et al., "Single chip OQPSK modem appropriate for wireless burst data communications," in *Proceedings of the International Conference on Signal Processing Applications and Technology*, Orlando, Fla, USA, November 1999.
- [5] W. Le, W. Zhu-gang, and X. Wei-ming, "High-accurate carrier acquisition based on maximum likelihood estimation of refined frequency," *Telecommunication Engineering*, vol. 53, no. 1, pp. 39–43, 2013.
- [6] F. Cheng and Q. Cheng, "The large sample performance of a maximum likelihood method for OFDM carrier frequency offset estimation," *Wireless Personal Communications*, vol. 72, no. 1, pp. 227–244, 2013.
- [7] S. Tao, L. Huijie, and L. Xuwen, "A kind of all digital phase locked loop used in the situation of tremendous frequency offset error and low signal to noise," *Journal of Electronics and Information Technology*, vol. 27, no. 8, 2005.
- [8] L. Baowei, C. Shijing, H. Yuanjie, and G. HongTao, "The design of QPSK carrier recovery loop in the environment of enormous frequency offset error and low signal to noise," *The Journal of Radio Engineering*, vol. 39, no. 4, pp. 51–54, 2009.
- [9] M. Kumm, H. Klingbeil, and P. Zipf, "An FPGA-based linear all-digital phase-locked loop," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 9, pp. 2487–2497, 2010.
- [10] E. B. Roland, *Phase-Locked Loop, Simulation, and Applications*, McGraw-Hill, New York, NY, USA, 5th edition, 2003.
- [11] S. K. Mitra, *Digital Signal Processing: A Computer-Based Approach*, McGraw-Hill, New York, NY, USA, 2001.
- [12] R. Jaffé and E. Rechten, "Design and performance of false-lock loops capable of near-optimum performance over a wide range of input signal and noise levels," *IRE Transactions on Information Theory*, vol. 1, no. 3, pp. 66–72, 1965.
- [13] A. J. Viterbi, *Principles of Coherent Communications*, chapter 2, McGraw-Hill, New York, NY, USA, 1966.
- [14] Z. Juesheng, Y. Zhengju, and P. wanxin, *Locked Phase Technology*, Xian University of Electronic Science and Technology Press, Xian, China, 1998.
- [15] A. I. Ahmed, S. H. Rahman, and O. A. Mohamed, "FPGA implementation and performance evaluation of a digital carrier synchronizer using different numerically controlled oscillators," in *Proceedings of the Canadian Conference on Electrical and Computer Engineering (CCECD '07)*, pp. 1243–1246, April 2007.
- [16] Xilinx LogiCORE IP DDS Compiler v4.0 Product Specification, Xilinx, April 2010.
- [17] E. D. Kaplan and C. J. Hegarty, *Understanding GPS: Principles and Applications*, Artech House, Norwood, Mass, USA, 2005.
- [18] P. K. Meher, J. Valls, T. B. Juang, and K. Maharatna, "50 years of CORDIC: algorithms, architectures, and applications," *IEEE Transactions on Circuits and Systems. I. Regular Papers*, vol. 56, no. 9, pp. 1893–1907, 2009.
- [19] E. Frantzeskakis and P. Koukoulas, "Phase domain maximum likelihood carrier recovery: framework and application in wireless TDMA systems," in *Proceedings of the 50th Fall IEEE VTS 50th Vehicular Technology Conference 'Gateway to 21st Century Communications Village' (VTC '99)*, pp. 2571–2575, IEEE, September 1999.
- [20] M. Luise and R. Reggiannini, "Carrier frequency recovery in all-digital modems for burst-mode transmissions," *IEEE Transactions on Communications*, vol. 43, no. 2–4, pp. 1169–1178, 1995.
- [21] L. Guojun and H. Decong, "The calculation of link margin of satellite communication in SKT," *The Journal of Space Electronic Technology*, vol. 1, no. 12, pp. 68–72, 2012.
- [22] W. Shiqi, W. Tingyong, and Z. Ningxing, *An Introduction to Satellite Communications*, Electronic Industry Press, Beijing, China, 2nd edition, 2006.
- [23] Y. Dangui, T. Ruijun, X. Min, and Z. Chengchang, "An optimal method for costas loop design based on FPGA," in *Proceedings of the 4th International Conference on Digital Manufacturing and Automation (ICDMA '13)*, pp. 175–179, Qingdao, China, June 2013.
- [24] H. Yuan, X. Hu, and J. Huang, "Design and implementation of costas loop based on FPGA," in *Proceedings of the 3rd IEEE Conference on Industrial Electronics and Applications (ICIEA '08)*, pp. 2383–2388, Singapore, June 2008.
- [25] K. He, L. Crockett, and R. Stewart, "Dynamic reconfiguration technologies based on FPGA in software defined radio system," *Journal of Signal Processing Systems*, vol. 69, no. 1, pp. 75–85, 2012.
- [26] T. Adiono and R. S. Purba, "Scalable pipelined CORDIC architecture design and implementation in FPGA," in *Proceedings of the International Conference on Electrical Engineering and Informatics (ICEEI '09)*, vol. 2, pp. 646–649, August 2009.

