Journal of VLSI Circuits and Systems, ISSN: 2582-1458 Vol. 6, No. 2, 2024 (pp. 1 -13)
RESEARCH ARTICLE
WWW.VLSIJOURNAL.COM

# Development of Synthesizable Filter-Centric Loop Filter Design for ADPLL Architecture in SoC

[1]G.Ravikumar, [2]D.Venkatareddy, [3]M. Asharani
[1]Research Scholar, Department of ECE, JNTU, Hyderabad, Telangana, India 500085
[2]Professor, Departme nt of ECE, MGIT, Gandipet, Hyderabad, , Telangana, India 500075
[3]Professor, Department of ECE, JNTU, Hyderabad, Telangana, India 500085

**KEYWORDS:**
ADPLL,
Digital Loop Filter,
DCO,
Zynq SoC, Jitter and
Phase Error.

**ARTICLE HISTORY:**
Received 15.04.2024
Accepted 20.05.2024
Published 11.06.2024

**DOI:**
https://doi.org/10.31838/jvcs/06.02.01

## Abstract

The emergence of all-digital phase-locked loop (ADPLL) filters marks a major advancement in the field of signal processing technology. These filters are engineered to align the phase and frequency of digital signals, ensuring accurate timing and accuracy in a wide range of applications. By using a digitally controlled oscillator (DCO) to adjust the frequency and compare it to a reference signal, ADPLL filters minimize phase differences, improve filtering efficiency, and reduce noise in digital communication systems. Due to their ability to adapt to different input signals and environmental factors, these filters are critical to modern electronic devices, especially in challenging applications such as wireless communications, data transmission, and clock synchronization. In this research work, the variable reset random walk filter is designed in ADPLL architecture. The complete ADPLL design is developed using Verilog HDL language and realized in Zynq 7000 all programmable SoC device. The resource and power utilization reports are compared with the existing ADPLL design. In other hands the Pk-to-Pk jitter and phase errors are compared, in all cases the proposed ADPLL design exhibits better performance compared to exiting ADPLL design.

**Author e-mail:** ravikumarg1308@gmail.com, dvenkatreddy_ece@mgit.ac.in, ashajntu1@jntuh.ac.in

**How to cite this article:** Ravikumar G, Venkatareddy D, Asharan M. Development of Synthesizable Filter-Centric Loop Filter Design for ADPLL Architecture in SoC, Journal of VLSI Circuits and System Vol. 6, No. 2, 2024 (pp. 1-13).

## Introduction

The emergence of all-digital phase-locked loop (ADPLL) filters represents a major advancement in the field of signal processing technology.[1] ADPLL filters are carefully designed to align the phase and frequency of digital signals, ensuring accurate timing and accuracy in a wide range of applications.[12.26,29] These filters use a digitally controlled oscillator (DCO) to change its frequency and compare it to a reference signal.[3] This minimizes phase differences, improves filtering efficiency, and reduces noise in digital communications systems. These filters are vital in modern electronic equipment because they improve signal quality and reliability. A key feature of ADPLL filters is their ability to adapt to different input signals and environmental factors. These filters use digital processing methods that, unlike standard analog PLL filters, allow different parameters to be modified in real time. Due to their flexibility, they are able to maintain stable performance under various operating environments, such as temperature, voltage, and signal strength.[4] ADPLL filters provide enhanced reliability and resilience in challenging applications such as wireless communications, data transmission and clock synchronization.[23,7]

Additionally, ADPLL filters offer significant benefits in terms of power efficiency and integration. They may also provide superior performance and low power consumption through the use of digital signal processing methods.[22] These filters are ideal for battery-operated devices and energy-efficient systems that prioritize low power usage. ADPLL filters can be easily created using standard CMOS technology, allowing smooth integration with other digital circuits on a single chip. The integration of these components reduces component count, board area occupied, and production-related expenses, making ADPLL filters an economical choice for large-scale electronic products. High-performance filters are crucial to the operation of ADPLL systems.[6] These filters are designed to meet stringent performance criteria such as minimal phase noise, good linearity and fast settling time .[24]

By using advanced digital signal processing methods, high-performance filters effectively reduce unwanted noise and interference while maintaining precise phase and frequency synchronization. These components are critical in modern communications and data processing applications where maintaining signal integrity is critical. High-performance filters use advanced algorithms and precise control mechanisms to ensure clear and stable output signals, which is critical for high-speed data transmission and wireless communication situations.[27]

The impact of different filter characteristics on the design and functionality of ADPLL is significant. Bandwidth, phase noise, settling time and linearity significantly affect the stability, accuracy and resilience of ADPLL in different operating conditions and applications.[8] The bandwidth of the filter has a significant impact on the speed and responsiveness of the ADPLL to track and lock onto the input signal. Narrow bandwidth filters provide more selectivity and noise rejection, but may result in slower response times and longer settling times. On the other hand, wider bandwidth. The filter provides faster locking and tracking capabilities, but may experience reduced noise rejection and increased interference susceptibility. Choosing the correct bandwidth for your ADPLL design is key to maintaining a balance between performance requirements and system dynamics.

The phase noise characteristics of the filter have a substantial impact on the signal quality and stability of ADPLL. Phase noise affects the integrity and reliability of synchronization signals and can be reduced by using filters with low phase noise levels. These filters are critical to reducing signal distortion, improving signal-to-noise ratio, and improving the overall performance of the system. Designers can improve signal synchronization in ADPLL by improving filter design parameters and using advanced noise reduction techniques to combat phase noise effects, allowing efficient operation under challenging conditions. The settling time of the filter affects the rate at which the ADPLL achieves phase and frequency lock. Fast settling times are critical for fast synchronization, reduced latency, and enabling real-time data processing and smooth connections in communication systems. Additionally, filter linearity is critical to maintaining signal integrity and reducing distortion in ADPLL systems. Filter nonlinearity can cause signal degradation and phase and frequency alignment errors, which can affect the efficiency of the ADPLL. Therefore, maintaining good linearity of the filter is critical to ensure reliable and accurate performance in a variety of applications. Optimizing filter characteristics is crucial to improve the performance, reliability, and efficiency of ADPLL systems in different signal processing tasks and applications. The organization of the paper as follows: The Section-II describes the Literature survey and the proposed ADPLL architecture explained in Section-III. The simulation results and Realization of SoC results are discussed in Section-IV.

## Literature

Huirem Bharat Meitei et al.[28] describe the creation, implementation and evaluation of a true random number generator (TRNG) using an ADPLL with a finite impulse response (FIR) filter as its digital loop filter. The TRNG is used on the Artix 7 FPGA board with the Xilinx Vivado 2015.2 design suite. Using the Keiser window method and with the help of MATLAB-FDA tool, the coefficients of the 3rd order broadcast low-pass digital FIR filter were calculated. The recommended ADPLL-based TRNG design uses an XOR corrector post-processing approach to remove biases in the sequence, resulting in unbiased random number generation, with a combined throughput of 200 Mbps for both designs.

Velamarthi Spandana et al. [10] proposed an innovative Hilbert-Huang based all-digital phase-locked loop (HH-ADPLL), with the purpose of achieving the locked state of the ADPLL. The input reference signal undergoes Hilbert-Huang transformation to extract the analytical components, resulting in rising and falling signals. After removing high-frequency components from the signal, the up/down counter generates borrow and carry signals. These signals are input to up-down counters to produce output signals. The phase detection module ensures that the input and output signals match, which causes the ADPLL to enter a locked state. HH-ADPLL was designed on Artix-7 FPGA and its effectiveness was confirmed by evaluating stability, phase error, power consumption, combined latency and performance improvements.[16]

Eugene Koskin et al.[11] explored interconnected oscillators used in generating clock signals for complex systems-on-chip. The oscillators are implemented as interconnected ADPLLs that operate as asynchronous control systems. The work explores the modeling, synchronization, and stability of both individual ADPLLs and linked ones. The stability domain is shown to be universal for extensive Cartesian networks and is linked to the domain of a single ADPLL. The network achieves synchronization with the reference signal in both frequency and phase within this stable zone. Hardware verification of Cartesian networks is performed to validate their conformity with theoretical results. The suggested architecture has the potential for several engineering and physics

applications, including as clock generation, distributed computing, beamforming, and other situations where accurate time synchronization is crucial for system efficiency.

Huirem Bharat Meitei et al.[30] presented an enhanced method that utilizes on-chip jitter and metastability state. The design has a 15-bit Linear Feedback Shift Register (LFSR) with ADPLL-based TRNG, using ring oscillators, flip-flops, ADPLL, 15-bit LFSR, and diverse physical components for creating entropy sources. The new approach minimizes complexity and power usage (0.072W) by using minimum FPGA hardware resources compared to conventional TRNG systems. This design exhibits higher performance compared to traditional TRNGs, which might lead to the creation of more efficient, low-power, and highly dependable TRNGs. Testing on an Artrix-7 FPGA evaluation board shows favorable results. The waveform and FFT pattern of the wave are captured using a digital storage oscilloscope (DSO). Statistical analysis of the output bit sequence from the design is performed using MATLAB. The NIST SP 800-22 assessment verifies the randomness and unpredictable nature of the TRNG bitstreams, highlighting the appropriateness of the proposed architecture for cryptographic purposes.

Mohd Ziauddin Jahangir et al.[13] discussed the creation and FPGA integration of a Sine-Wave Direct Digital Synthesizer-All-Digital Phase-Locked Loops designed for resonant sensing applications, namely those related to mechanical structures.[14] The PLL is designed to detect changes in the resonant frequency of a mechanical structure at various spatial orientations, with an expected resonant frequency below 10 KHz. Most CMOS PLL ICs are not suited for this application due to their much higher operating frequencies. A digital PLL solution is preferred for its tremendous configurability. The paper details the structure and construction of two separate sine wave DDS ADPLLs. An FPGA-based Type-1 sine wave DDS ADPLL is implemented utilizing SPI ADCs and DACs for communication. The article also explains the important design factors for implementing the suggested ADPLL hardware. The ADPLL that was put into operation exhibits perfect locking behavior at a central frequency of 7KHz.

E. B. Priyanka et al.[27] aims to combine an 8-GHz voltage-controlled oscillator (VCO) with a frequency tripler to produce a 24-GHz local oscillator. Using a recycled architecture for stacking the VCO and tripler allows for power reductions in this integration. The circuit is implemented in a 0.18 µm CMOS technology and has a chip size of 0.7 mm × 0.8 mm. Experimental data shows that the frequency tuning range (FTR) of the VCO changes from 7.06 to 8.33 GHz when the tuning voltage increases from 0 to 2 V.

Kalpana Kasilingam et al.[15] created a more advanced phase-locked loop (PLL) using an upgraded version of the phase-frequency detector that includes a multi-band flexible frequency divider. This results in higher frequency resolution, spectral quality and output signal fidelity. Previous PLL designs suffered from significant timing jitter problems, caused by an imbalance in the frequency transfer function of the voltage-controlled oscillator and noise caused by supply voltage changes. To address these issues, new phase frequency detector (PFD) design strategies are recommended by reducing PLL lock time and reducing phase jitter. This unique approach results in reductions in transistor count, power consumption, propagation delay and overall size compared to traditional static PFDs. The addition of a forward loop voltage controlled oscillator improves the resolution of frequency and phase variation errors by ensuring balanced driving force ratios in the feedforward and feedback paths.

Rachana Ahirwar et al.[31] emphasized the use of Cadence Virtuoso analog design platform and SCL 180nm manufacturing technology to create PLL systems. A single-ended voltage controlled oscillator was chosen because of its beneficial properties such as small chip size, low power consumption, and wide frequency range. The simulation results were confirmed through the Specter simulator in the Cadence Virtuoso tool. Abdelrahman S. Moustafa et al.[17] introduced an automated method for creating and optimizing ADPLL systems. This program is designed to determine the coefficients of a digital loop filter (DLF) using analytical noise models and phase noise simulation results for various components of the ADPLL. Use evolutionary optimization methods to fine-tune the DLF coefficients to obtain the best lock time and phase noise performance for the ADPLL. The tool generates simulation results of ADPLL designs that match the required system requirements.

Chia-Chen Chang et al.[18] demonstrated an ADPLL with adaptive high-order filters. This ADPLL variant allows selection of the loop filter order from first to third order via IIR filter modification, helping to optimize system performance and reduce input noise. The work also examined the convergence of spurious tones toward the main tone in high-order ADPLL settings. The device was created using TSMC's 40nm GP 1P10M CMOS manufacturing technology. Ke Wu et al.[19] outlined an adaptive Kalman filter (AKF) digital phase detector designed to reduce output noise in an a ADPLL. After mathematical modeling and noise source analysis, an

adaptive Kalman filter was customized for the ADPLL model and added after the phase detector to enhance the noise reduction effect. The AKF optimally adjusts its parameters based on the statistical properties of the noise to effectively act as a filter. MATLAB simulation confirmed the accuracy and reliability of the proposed design.

Mohd Ziauddin Jahangir et al.[20] proposed a new method to improve the frequency resolution of digitally controlled oscillators (DCOs) without generating frequency spikes. This study introduces a new hybrid DCO architecture, designed to increase resolution. Hybrid DCO architecture combines digital and analog control methods for frequency adjustment. Digital control inputs allow for extensive frequency modification, while analog control inputs allow for precise frequency adjustments. Experimental evidence shows that combining a hybrid DCO with Delta-Sigma Modulation (DSM) and an analog low-pass filter (LPF) can effectively improve resolution without causing spectral spikes. Two hybrid digitally controlled ring oscillators (DCROs) were created using 90nm CMOS technology as a proof of concept and their cycle jitter performance was analyzed.

## Proposed Model

The main goal of the research is to design ADPLL which achieves a digital lock phase, based on a low-frequency clock source, which produces high-frequency clocks, and achieves the expected effect of high-frequency clock output phase locking, accounting for 1:1. The design is based on the Verilog code. The first stage achieves a digital lock link to achieve the effect of output locking, and the second stage carries out multi-frequency realization. The proposed ADPLL block diagram is depicted in Figure 1.

Figure 1 shows the structure of an ADPLL with modified Filter Block. It is mainly composed of 4 parts: digital phase detector (DPD), digital loop filter (DLF), Phase Controller, and frequency divider. ADPLL is a circuit structure that controls the system through phase feedback. The phase error signal between the input signal and the local clock output signal is sent to the digital loop filter, and the relative error is smoothed and filtered to generate control signals carry and row. The digital oscillator adjusts the feedback according to the control signal. The phase of the output signalgradually tracks the phase of the input signal , and finally reaches locking.

## Phase Detector

In an ADPLL, the phase detector is a key component responsible for measuring the phase difference between the input signal and the feedback signal and generating a corresponding error signal, which is then used to adjust the DCO to reduce the phase difference between the input signal and the output signal to achieve phase locking. Phase detectors can be implemented in a number of different ways, the most common of which include XOR gate-based phase detectors and edge-triggered based phase detectors. The XOR gate phase detector generates an error signal by comparing the phase difference between the input signal and the feedback signal. When the two signals are completely synchronized, the output is zero; when there is a phase difference, the duty cycle of the output signal will reflect this phase difference. The edge-triggered phase detector is triggered when the rising edge or falling edge of the input signal or feedback signal occurs, and generates the corresponding control signal.

In ADPLL design, the performance of the phase detector has a direct impact on the overall performance of the system. A high-performance phase detector should be able to accurately measure small phase differences and maintain stable output under different operating conditions. In addition, in order to reduce the impact

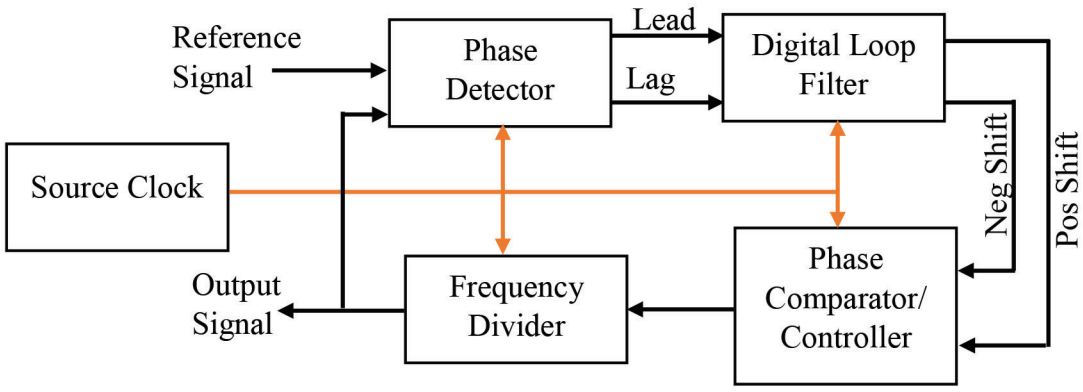


**Fig. 1: Proposed ADPLL Block Diagram**

of noise, the phase detector usually needs to be used in conjunction with a DLF, which can smooth the output of the phase detector, thereby improving the stability and anti-interference ability of the system. In practical applications, the phase detector of ADPLL not only needs to have high accuracy and stability, but also needs to take into account the implementation complexity and power consumption. With the development of integrated circuit technology, FPGA-based ADPLL design provides powerful flexibility and computing power to implement complex phase detection algorithms, allowing ADPLL to be used in a wide range of application fields, such as wireless communications and high-precision clocks.[24] Synchronization, etc., providing high-performance solutions.

## Digital Loop Filter

In this research work, a variable reset random walk filter is designed as a loop filter module. In ADPLL, the variable reset random walk filter module is an advanced digital loop filter design designed to optimize the performance of phase locked loops, especially in low signal-to-noise ratio environments or when facing rapidly changing frequency conditions. This filter responds to random fluctuations in the input signal by dynamically adjusting its parameters, thereby improving the phase tracking capability and system stability of ADPLL. The core function of the variable reset random walk filter module is to not only generate a correction signal to adjust the output of the DCO when a phase difference is detected, but also to dynamically adjust the filter response based on changes in the input signal characteristic. This dynamic adjustment capability allows the filter to effectively suppress frequency jitter caused by random fluctuations in the input signal while maintaining a locked state. The proposed variable reset random walk filter is depicted in Figure 2.

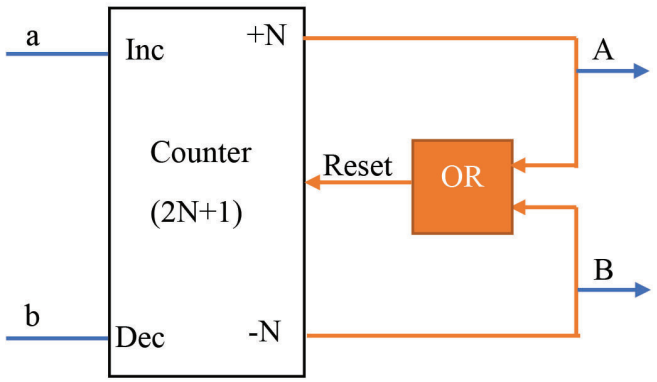


**Fig. 2: Proposed Variable Reset Random Walk Filter**

Variable Reset Random Walk Filter Module is a circular filter module. The sub-module of Random Walk Filter is realizing Random wandering sequence filter. The main body of the random wandering sequence filter is a reversible counter. When there is a forward pulse loser to the *UP* end, the counter is counted upwards, and when there is a lag pulse input to the *DN* end, the counter is counted down. If the number of advanced pulses exceeding the lag reaches the counting capacity *N*, an early pulse is exported at the end of *+N*, and the counter is reset. Conversely, a post-push pulse is exported at the *-N* end, and the counter is reset. Before the circuit is locked, the phasing device continuously exports the forward or lagging pulse. After the upper counter or lower counter reaches the full state, it exports the early pulse or pushed back pulse. Under these two pulses, the circuit gradually enters the locking state. When the loop enters a locked state, the ultra-front or lag pulses caused by noise are random, and the probability of occurrence is basically equal, and there will not be many consecutive ultra-front or lagging pulses, so their difference reaches the counting capacity *N*. Very small, so that the interference effect of noise on the loop can be reduced.

The Proposed filter can automatically adjust its internal parameters, such as the bandwidth and damping coefficient of the filter, according to the stability and noise level of the input signal to adapt to different operating conditions. When faced with rapid frequency changes, the filter can quickly adjust its output, reduce phase tracking delays, and quickly return to the locked state. By optimizing the filter design, the jitter of the phase-locked loop output frequency caused by input signal noise can be significantly reduced and the overall stability of the system can be improved.

The variable reset random walk filter module has significant advantages in processing low signal-to-noise ratio signals or in highly dynamic environments, ensuring that the ADPLL system maintains high-performance operation under complex conditions. This makes this module particularly important in applications such as wireless communications, satellite navigation, and high-precision clock synchronization,[2] where system performance relies heavily on high stability and fast response phase-locked loop designs.[5,9] The variable reset random walk filter module provides ADPLL designs with an effective solution to challenges caused by input signal uncertainty. Through its dynamically adjusted and optimized filtering capabilities, it not only improves the robustness of the system, but also enhances the performance of the phase-locked loop, enabling it to play a key role in a wider range of application scenarios.

As digital phase locking technology continues to advance, the design and application of this filter module will continue to be optimized and expanded to meet higher performance requirements in the future.

### Phase Controller

In the ADPLL system, the phase controller is responsible for adjusting the DCO according to the error signal provided by the phase detector to achieve phase synchronization of the input signal and the output signal. Its main purpose is to reduce or eliminate the phase error between the input signal and the feedback signal by adjusting the control word of the DCO, thereby achieving stable phase locking. The working principle of the phase controller is based on the phase error signal, which is generated by the phase detector and reflects the phase deviation of the current feedback signal relative to the input reference signal. The phase controller dynamically adjusts the frequency and phase output of the DCO based on this phase error signal, so that the output signal of the phase-locked loop can track the phase change of the input signal.

In ADPLL design, the design of the phase controller is crucial because it directly affects the locking time, stability, and frequency and phase tracking accuracy of the system. To optimize these performance metrics, phase controllers are often used in conjunction with loop filters, which smooth the control signal and reduce noise-induced frequency and phase jitter. In addition, to cope with different operating conditions and application requirements, the phase controller may need to support different control strategies, such as linear control, nonlinear control, or look-up table-based control. In some high-performance ADPLL designs, the phase controller may also implement more complex control algorithms, such as proportional-integral-derivative (PID) control, to further improve the dynamic performance and stability of the system. With the development of modern integrated circuit technology, ADPLL design based on FPGA or ASIC provides a powerful platform for efficient and flexible phase control, enabling ADPLL to meet a variety of application scenarios including wireless communications, precise clock synchronization, and signal modulation. need. Therefore, the design and implementation of the phase controller is a key link in the design of the ADPLL system, which is of great significance to ensuring the overall performance and reliability of the system.

### 3.4 Frequency Divider

In the ADPLL system, the function of the frequency divider is to divide the high-frequency signal generated by the DCO so that it can be compared with the input signal or a reference signal called phase comparison. The frequency divider directly affects the performance of the ADPLL system, including the operating frequency range, phase noise characteristics and locking accuracy of the phase-locked loop. The frequency divider implements the frequency division function by counting the input signal. In ADPLL, the frequency divider is usually placed between the DCO and the phase detector. By properly allocating the period of the DCO output signal, the frequency of the signal output by the frequency divider is reduced so that the frequency of the output signal becomes an integer fraction of the original signal. In this way, even if the frequency of the DCO is higher, a lower frequency signal can be obtained after frequency division to match the frequency of the reference signal for phase comparison.

In ADPLL design, different types of frequency dividers can be used according to application requirements, including fixed frequency dividers and programmable frequency dividers. The frequency division ratio of the fixed frequency divider is fixed and is suitable for scenarios where the output frequency requirement is relatively stable. The programmable frequency divider provides higher flexibility and can dynamically adjust the frequency division ratio according to actual needs to adapt to a wider frequency range and different application scenarios. In the ADPLL system, the frequency divider is not only used to down-convert the DCO's high-frequency signal for effective phase comparison with the low-frequency reference signal, but also affects the phase locking range and accuracy of the system. By precisely controlling the frequency division ratio, the frequency response of the phase-locked loop can be effectively adjusted to achieve precise frequency locking and stable phase synchronization.

## Experimental Results

This section describes the Experimental results of Proposed ADPLL with help of simulation waveforms. The structure of the proposed ADPLL is divided into four modules, Top-level module which configure all submodules and the sub modules are Phase comparator, Variable Reset Random Walk Filter and Frequency divider. The system works synchronously under a unified system clock. The Schematic of Proposed ADLL is depicted in Figure 3.

The Top module, which is responsible for combining the various sub-modules and declaring the signal interface of the entire system. Phase comparator/ detector is used as phase identifier module, first use the '*D*' trigger

to synchronize the input signal into the system's clock domain; second, the statistical input signal cycle is used to generate oscillating signals; third, the input signal is compared with the feedback signal, The feedback signal is ahead of the input signal and the Lead is high level, on the contrary, Lag is placed as high level when the feedback signal lags, and the effective duration of both signals is a system clock cycle. The simulation result of Phase comparator is depicted in Figure 4.

The system clock is set to 50MHz, and the input signal frequency is 500KHz. The input signal is uncertain, that is, a random delay is added to the simulation, which is set to 3287ns. The random delay can be seen in Figure 5.

The main function is to divide the frequency based on the system clock. The system clock depends on the FPGA hardware model, which generally requires a higher system clock. Based on the number of clock cycles for input signals calculated by the above phase detector module, the number of clock cycles for multi-frequency signals is calculated to produce synchronous oscillation signals. Due to the delay of the system, the oscillating signal generated at the beginning is not synchronized with the input signal, and it is processed by the phasing device to gradually reach the synchronous locking state. The simulation waveform of the proposed ADPLL is shown in Figure 6.

In Figure 6, MainClock is the system clock, SignalIn is the input signal, SignalOut is the output signal, and Lead and Lag are the phase leading and lag signals of the phase indicator output, respectively. Positive and Negative correspond to the above two The signal is exported after the circular filter. Lock is an output lock signal, which locks the high-power peacetime loop. As can be seen from the simulation waveform graph, the locking time is 152300ns, minus the input delay of 3287ns added at the beginning, and the actual locking time is 149013ns,

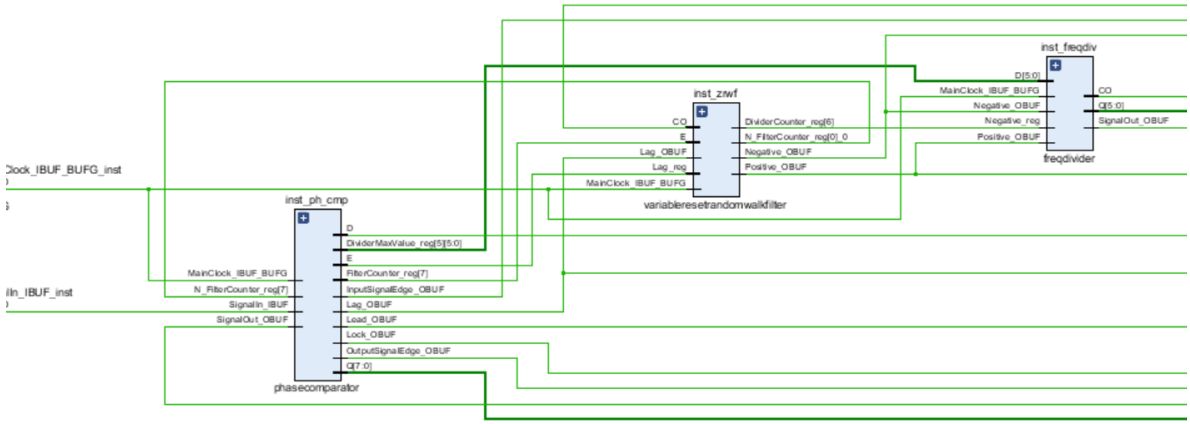


**Fig. 3: Schematic of Proposed ADPLL**

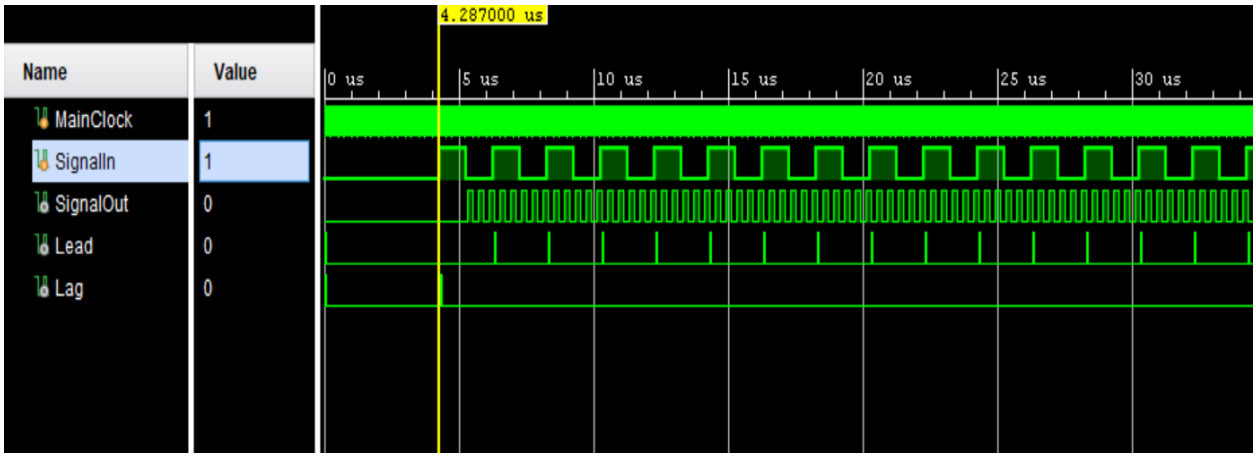


**Fig. 4: Simulation Result of Phase comparator**

or 149µs. After the locking, the output and the input signal are the same frequency. By changing the Divider parameter in the top file, Divider Multiple can modify the multiplier of the chain output, which can achieve odd frequency and even frequency. As shown in the Figure 7 and 8, they are based on the 500KHz input signal to produce 4 times and 20 times.

From Figure 7 and 8, it is evident that this design implements a full-digital lock-in loop that can produce high-frequency clocks based on low-frequency clock sources, and achieves the expected effect of high-frequency clock output phase locking and accounting for 1:1. Through this design, the lock-in loop can be achieved on the FPGA platform, locking the unknown lower-frequency input signal, and generating any multi-frequency (frequency of the same frequency is lower than the system clock frequency) output signal. The proposed ADPLL is realized in Zynq-7000 All programmable SoC device. In order to realize the proposed ADPLL, the Verilog source code is synthesized and Netlist is generated in the Xilinx Vivado tool. The results obtained from SoC device is shown in Figure 9 and 10.

The results of Zynq SoC device is observed in Integrated Logic Analyzer (ILA) core. The Figure 9 represents the

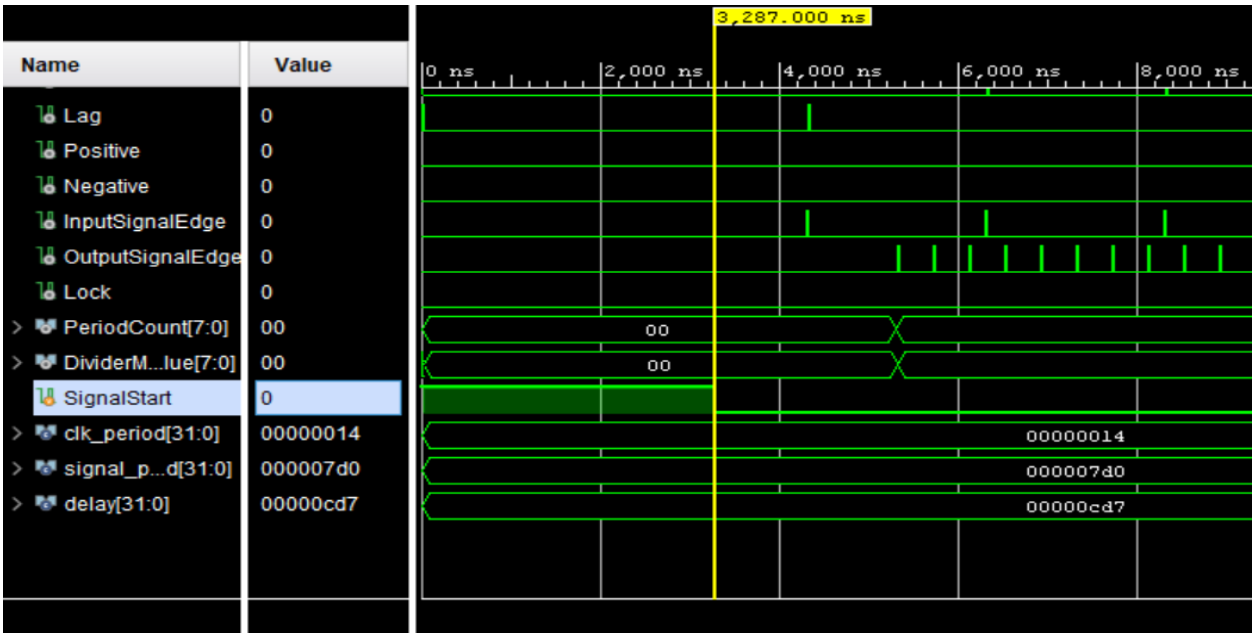


**Fig. 5: The simulation waveform of Signal Start condition**

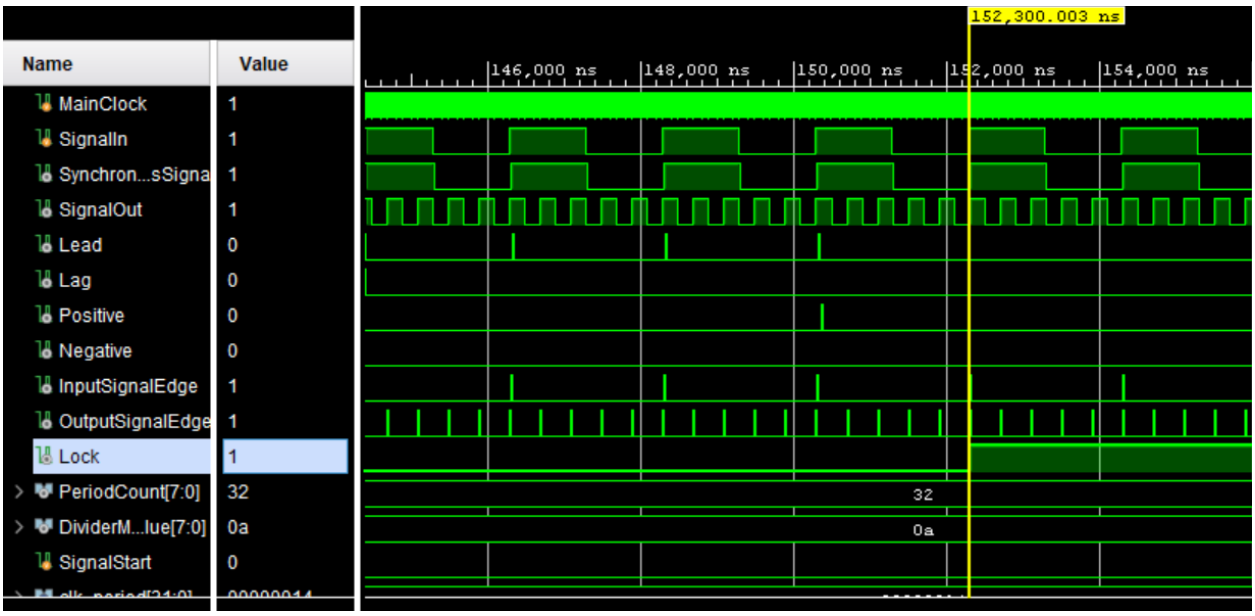


**Fig. 6: Simulation Results of Proposed ADPLL**

lag signal is asserted which indicates the phase of Output clock signal is lag when compared to Reference clock signal. The lead and lag signal assertion is done by the phase comparator and the output clock signal is move shift left or right based on the output of the phase comparator. The Figure 10 indicates the locking status of the Reference clock and Output clock signals. The corresponding Output and Reference clocks configuration parameters and jitter metrics are reporting in Table 1.

This Figure 11 shows the resource utilization of a proposed ADPLL design after synthesis (Post-Synthesis) and implementation (Post-Implementation). The figure 11 lists several different types of FPGA resources and

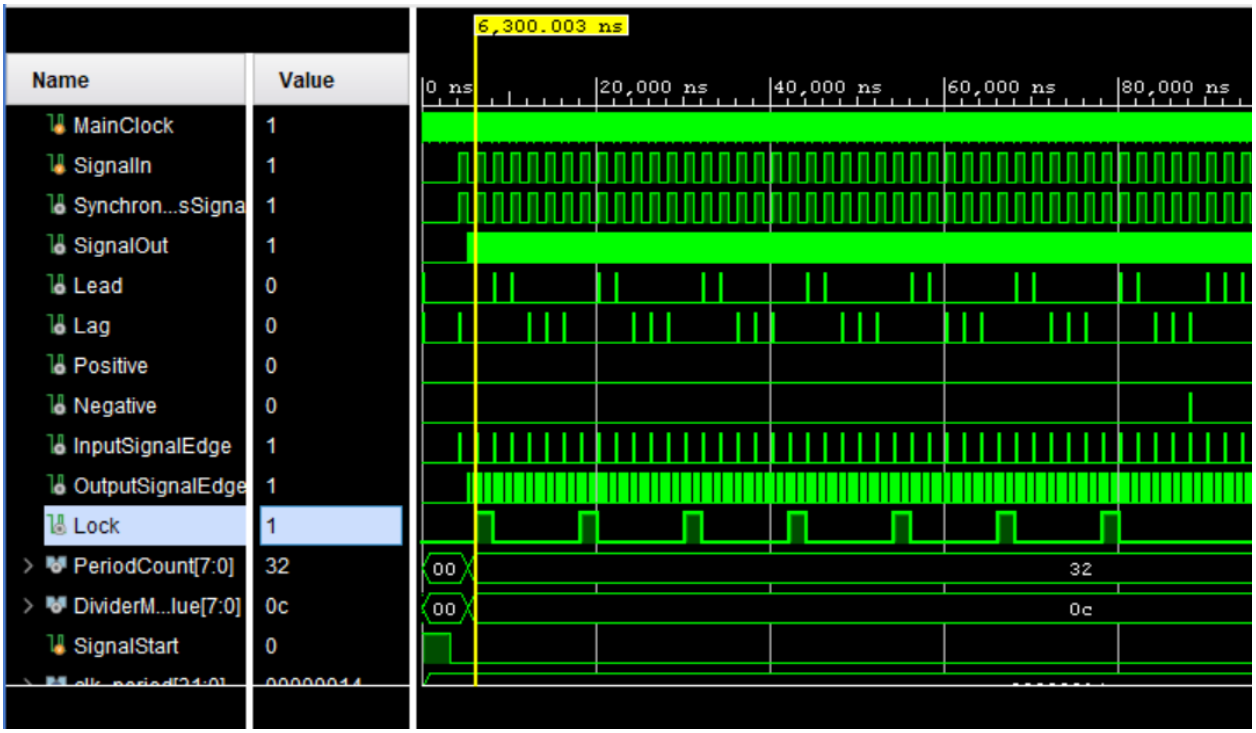


**Fig. 7: Simulation Result of ADPLL with Divider parameter=4.**

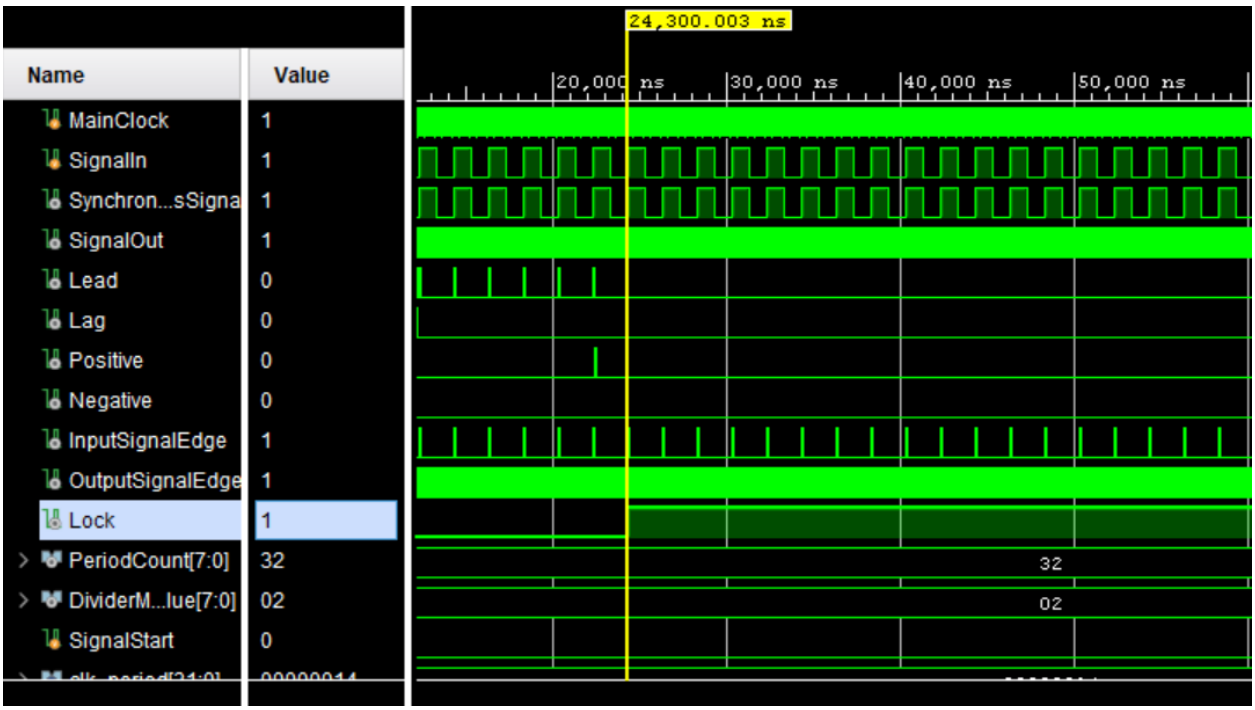


**Fig. 8: Simulation Result of ADPLL with Divider parameter=20.**

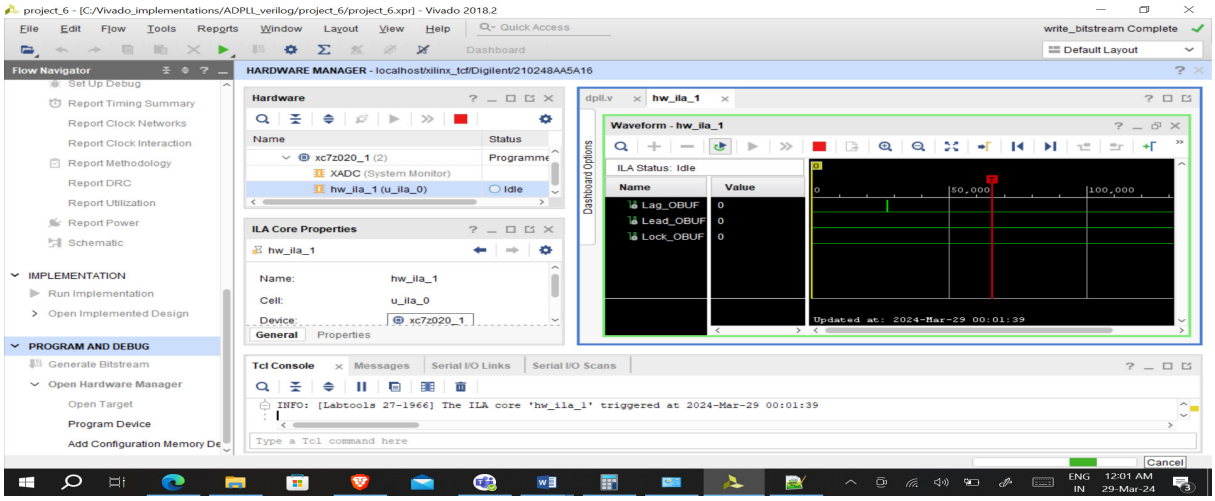


**Fig. 9: The Lag Signal Asserted in ILA core.**

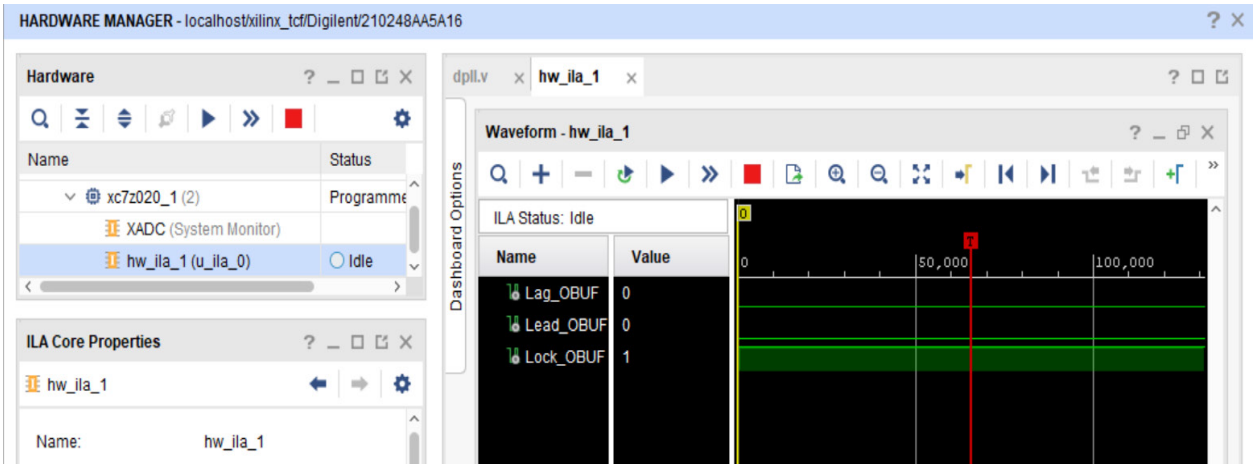


**Fig. 10: The Lock signal is Asserted in ILA core**

their usage percentages. LUT (look-up table): The utilization rate after synthesis is 3%. This is the main resource used to implement logic functions inside the FPGA. A utilization rate of 3% indicates that only a small part of the logic resources in the design are used. LUTRAM (Lookup Table Random Access Memory): The combined utilization rate is 1%. LUTRAM is a LUT that can be used as RAM. A utilization rate of 1% means that the use of such storage resources is also very low. FF (flip-flop): The utilization rate after synthesis is 2%. Flip-flops are used to store bit information and form the basis of sequential logic. A utilization rate of 2% means less timing resources are occupied. BRAM (Block Random Access Memory): The combined utilization rate is 11%. BRAM is usually used to achieve greater data storage requirements. The 11% utilization rate indicates a certain amount of data storage requirements. IO (input/output): The combined utilization rate is 6%, indicating that the design uses some IO resources, but not all are occupied. BUFG (global buffer): The comprehensive utilization rate is 6%. The global buffer is usually used to distribute clock signals or other global signals. The 6% utilization rate indicates that these signals have a certain distribution

**Table 1: Clock Configuration and Jitter Values**

| S.No | Parameter | Metric value |
|---|---|---|
| 1 | Reference Clock (MHz) | 100 |
| 2 | Programmable Divider | 2.735 |
| 3 | Output Clock 1 (MHz) | 500 |
| 4 | Output Clock 2 (Hz) | 50 |
| 5 | Lock-in Time (µs) | 149µs |
| 6 | Jitter | 0.010 |
| 7 | Pk-to-Pk Jitter (ps) | 86.824 |
| 8 | Phase Error (ps) | 87.466 |

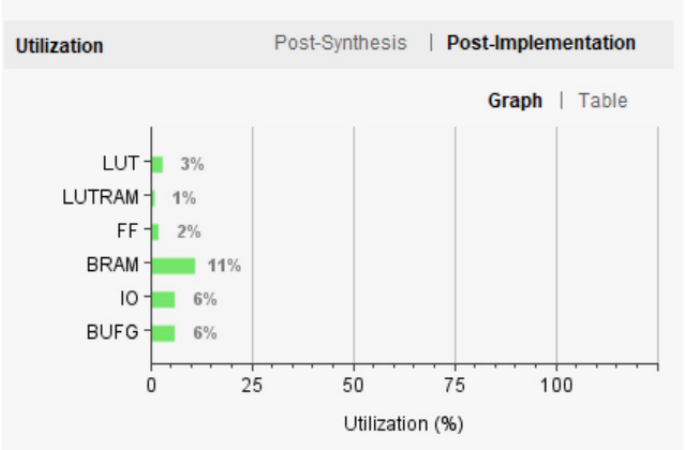


**Fig. 11. Resource Utilization Report**

but have not reached resource saturation. This resource utilization report is often used to evaluate whether a design effectively utilizes the FPGA's resources.

The Figure 12 shows power utilization report indicates the power consumption of a SoC device. Total On-Chip Power: 11.4 mW, this refers to the total power consumed when the chip is running. Junction Temperature: 26.3 degrees Celsius, the temperature inside the chip. Thermal Margin: 58.7 degrees Celsius (corresponding to 4.9 watts), indicating the additional temperature rise that the chip can withstand without exceeding the maximum operating temperature. Effective JA: 11.5°C/W, which is the temperature increase caused by each watt of power under normal working conditions of the chip. The report shows that the total power consumption of the chip is relatively low and the junction temperature is controlled within a safe range.

When comparing the existing ADPLL with the proposed ADPLL design, a series of significant performance improvements can be observed. First, the total number of slices is reduced from 4310 in the existing ADPLL to 137 in the proposed ADPLL, which shows that the new design greatly reduces the usage of hardware resources. Similarly, the number of slice lookup tables (LUTs) is also

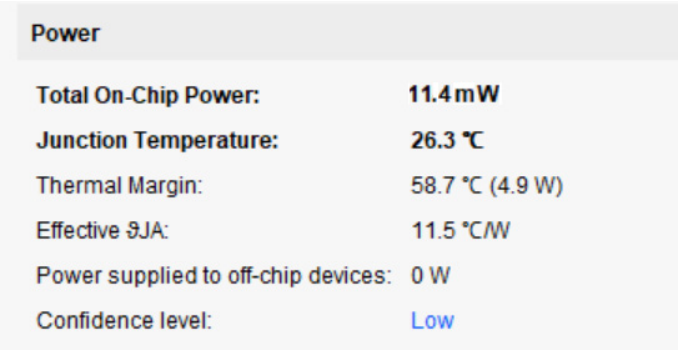


**Fig. 12: The Power Resource Utilization Report**

reduced from 13772 to 137, further proving the advantages of the proposed design in terms of resource utilization efficiency. In terms of slice registers, the number of registers in the existing ADPLL is also significantly reduced from 13,970 to 85 in the proposed ADPLL. This change also shows the effectiveness of the proposed solution in simplifying hardware design. Regarding the use of block RAMs (Block RAMs), the proposed ADPLL design reduces their number to zero, thereby further reducing the demand for memory resources.

In terms of power consumption, the proposed ADPLL only requires 11.4 milliwatts, which significantly reduces energy consumption compared to the existing ADPLL's 28 milliwatts, which is particularly important for applications that require low-power operation. In terms of peak-to-peak jitter (Pk-to-Pk Jitter) performance, the proposed ADPLL performs better in jitter control, reducing it from the existing 128.65 picoseconds to 86.824 picoseconds. The phase error also improved from 132.943 picoseconds of the existing ADPLL to 87.466 picoseconds of the proposed ADPLL, showing the effectiveness of the new design in improving signal accuracy and system stability. In summary, the proposed ADPLL design exhibits significant improvements in multiple aspects such as hardware resource utilization, power consumption, jitter control, and phase accuracy. These advantages make it ideal for many high-precision and low-power applications.

**Table 2: The Performance Comparison Results**

| S.No | Parameter | Existing ADPLL [21] | Proposed ADPLL |
|---|---|---|---|
| 1 | Total Slices | 4310 | 137 |
| 2 | Slice LUTs | 13772 | 137 |
| 3 | Slice Registers | 13970 | 85 |
| 4 | Block RAMs | 10 | 0 |
| 5 | Power Consumption | 28mW | 11.4mW |
| 6 | Pk-to-Pk Jitter (ps) | 128.65 | 86.824 |
| 7 | Phase Error (ps) | 132.943 | 87.466 |

## Conclusion

A synthesized filter center loop filter design is studied and developed for ADPLL architecture in SoC. By analyzing the impact of different filter characteristics, the study highlighted the significant impact of bandwidth, phase noise, settling time and linearity on ADPLL performance. In order to optimize the performance index, a variable reset random walk filter is proposed, and its role and advantages in the ADPLL system are described in detail. The design of the phase controller and frequency divider is also considered to be a key factor affecting the system locking time, stability and frequency tracking accuracy. Experimental results demonstrate the effectiveness and superiority of the proposed ADPLL design, especially in terms of hardware resource utilization, power consumption, phase jitter control, and phase accuracy. Therefore, this research provides a high-performance, low-power solution for ADPLL systems, especially suitable for demanding communication and synchronization applications. As digital phase locking technology continues to advance, it is expected that the design and application of this filter module will continue to be optimized and expanded to meet future higher performance needs.

## References


[1] Bhattacharyya, Sabyasachi, Aradhana Misra, and Kandarpa Kumar Sarma. "DPLLs in wireless communication: current state and trends—a literature survey." *Journal of The Institution of Engineers (India): Series B* 100 (2019): 627-647.

[2] Tu, Z., Zhou, H., Li, K., & Li, G. (2019). DCTG: Degree Constrained Topology Generation Algorithm for Software-defined Satellite Network. *Journal of Internet Services and Information Security*, *9*(4), 49-58.

[3] Lo, Yu-Lung, Hsi-Hua Wang, Yu-Hsin Li, Fang-Yu Fan, Chun-Yen Yu, and Jen-Chieh Liu. "A low-jitter all-digital PLL with high-linearity DCO." *Microsystem Technologies* 27 (2021): 1347-1357.

[4] Abdelatty, Omar, Xing Chen, Abdullah Alghaihab, and David Wentzloff. "Bluetooth Communication Leveraging Ultra-Low Power Radio Design." *Journal of Sensor and Actuator Networks* 10, no. 2 (2021): 31.

[5] Shaik, S. (2020). A coplanar wave guide fed compact antenna for navigational applications. *National Journal of Antennas and Propagation (NJAP)*, *2*(1), 7-12.

[6] Ambikapathy, A., Deepak Kumar, Kshitij Bhatnagar, Aman Srivastava, Abu Bakar, and Gajendra Singh. "Identification of Suitable PLL Technique for Grid-Connected PV System." In *Advances in Power Systems and Energy Management: Select Proceedings of ETAEERE 2020*, pp. 343-352. Springer Singapore, 2021.

[7] Ram, A., & Chakraborty, S. K. (2024). Analysis of Software-Defined Networking (SDN) Performance in Wired and Wireless Networks Across Various Topologies, Including Single, Linear, and Tree Structures. *Indian Journal of Information Sources and Services*, *14*(1), 39-50.

[8] Chen, Xing, and David D. Wentzloff. "Design Considerations of Frequency Modulated Ultralow Power Transmitter." In *Handbook of Biochips: Integrated Circuits and Systems for Biology and Medicine*, pp. 815-844. New York, NY: Springer New York, 2022.

[9] Cho, W., Park, Y., Sur, C., & Rhee, K.H. (2013). An Improved Privacy-Preserving Navigation Protocol in {VANET} s. Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 4(4), 80-92.

[10] Spandana, Velamarthi, and Chandra Sekhar Paidimarry. "A Novel Design of Hilbert Huang Based All Digital Phase Locked Loop Using FPGA."

[11] Koskin, Eugene, Pierre Bisiaux, Dimitri Galayko, and Elena Blokhina. "All digital phase-locked loop networks for clock generation and distribution: Network stability, convergence and performance." *IEEE Transactions on Circuits and Systems I: Regular Papers* 68, no. 1 (2020): 406-415.

[12] Yağız, E., Ozyilmaz, G., & Ozyilmaz, A. T. (2022). Optimization of graphite-mineral oil ratio with response surface methodology in glucose oxidase-based carbon paste electrode design. Natural and Engineering Sciences, 7(1), 22-33.

[13] Jahangir, Mohd Ziauddin, P. Chandra Sekhar, Md Sikander, and J. Vamshi Krishna. "Design and Implementation of FPGA based DDS-ADPLL for Resonant Frequency Tracking in Sensors." In *2022 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, pp. 1-6. IEEE, 2022.

[14] Bašić, Z. (2018). Analysis of the Application of Roadway Constructions in the Local Network Roads. Archives for Technical Sciences, 2(19), 29-34.

[15] Kasilingam, Kalpana, Paulchamy Balaiyah, Stephen Jeswinde Nuagah, and Piyush Kumar Shukla. "Design of a high-performance advanced phase locked loop with high stability external loop filter." *IET Circuits, Devices & Systems* 17, no. 1 (2023): 1-12.

[16] Charabi, Y., & Farhani, S. (2023). FPGA Application: Realization of IIR filter based Architecture. Journal of VLSI Circuits and Systems, 5(2), 29-35.

[17] Moustafa, Abdelrahman S., Hesham Omran, and Khaled Sharaf. "Automatic all-digital phase-locked loop system design and optimization tool." In *2020 15th International Conference on Computer Engineering and Systems (ICCES)*, pp. 1-6. IEEE, 2020.

[18] Chang, Chia-Chen, Yu-Tung Chin, Hossameldin A. Ibrahim, Kang Yu Chang, and Shyh-Jye Jou. "A Low-Jitter ADPLL with Adaptive High-Order Loop Filter and Fine Grain Varactor Based DCO." In *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1-5. IEEE, 2021.

[19] Wu, Ke, Chenyue Shi, Yuchen Li, and Changzhan Gu. "An adaptive Kalman filter based digital phase detector for all digital phase-locked loop." In *2021 IEEE MTT-S International Wireless Symposium (IWS)*, pp. 1-3. IEEE, 2021.

[20] Jahangir, Mohd Ziauddin, and Paidimarry Chandra Sekhar. "Design of novel hybrid-digitally controlled oscillator for ADPLL." *Memories-Materials, Devices, Circuits and Systems* 4 (2023): 100052.

[21] Radhapuram, Saichandrateja, Takuya Yoshihara, and Toshimasa Matsuoka. "Design and emulation of all-digital phase-locked loop on FPGA." *Electronics* 8, no. 11 (2019): 1307.

[22] Chen, Xing, Jacob Breiholz, Farah B. Yahya, Christopher J. Lukas, Hun-Seok Kim, Benton H. Calhoun, and David D. Wentzloff. "Analysis and design of an ultra-low-power bluetooth low-energy transmitter with ring oscillator-based ADPLL and 4$\times $ frequency edge combiner." *IEEE Journal of Solid-State Circuits* 54, no. 5 (2019): 1339-1350.

[23] Surender, S., Venkatachalam, K., & Gowrishankar, V. (2017). 6-bit, 180nm Digital to Analog Converter (DAC) Using Tanner EDA Tool for Low Power Applications. International Journal of Communication and Computer Technologies (IJCCTS), 5(2), 82-88.

[24] Radhapuram, Saichandrateja, Takuya Yoshihara, and Toshimasa Matsuoka. "Design and emulation of all-digital phase-locked loop on FPGA." *Electronics* 8, no. 11 (2019): 1307.

[25] Kang, J., Kim, J., & Sohn, M. M. (2019). Supervised learning-based Lifetime Extension of Wireless Sensor Network Nodes. *Journal of Internet Services and Information Security*, *9*(4), 59-67.

[26] Salarpour, Mahdi, Forouhar Farzaneh, and Robert Bogdan Staszewski. "Synchronization-phase alignment of all-digital phase-locked loop chips for a 60-GHz MIMO transmitter and evaluation of phase noise effects." *IEEE Transactions on Microwave Theory and Techniques* 67, no. 7 (2019): 3187-3199.

[27] Kangbin, Y. (2010). Guest Editorial: Advances in Trust Management. Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 1(4), 1-2.

[28] Meitei, Huirem Bharat, and Manoj Kumar. "FPGA implantations of TRNG architecture using ADPLL based on FIR filter as a loop filter." *SN Applied Sciences* 4, no. 4 (2022): 96.

[29] Gökhan, N. U. R., BARIŞ, B. N., Levent, B., SAZAKLIOĞLU, B. S., & Elvan, A. K. (2023). BUSER Transcutaneous Electric Nerve Stimulator Device Design. *Natural and Engineering Sciences*, *8*(1), 18-30.

[30] Meitei, Huirem Bharat, and Manoj Kumar. "FPGA-Based True Random Number Generator Architecture Using 15-Bit LFSR and ADPLL." In *International Conference on Science, Technology and Engineering*, pp. 275-288. Singapore: Springer Nature Singapore, 2023.

[31] Ahirwar, Rachana, Hemant Kumar Shankhwar, Gaurav Kaushal, Manisha Pattanaik, and Pankaj Srivastava. "Design High Frequency Phase Locked Loop Using Single Ended VCO for High Speed Applications." In *2022 IEEE Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI)*, pp. 1-6. IEEE, 2022.

[32] Priyanka, E. B., S. Thangavel, and V. G. Pratheep. "Enhanced digital synthesized phase locked loop with high frequency compensation and clock generation." *Sensing and Imaging* 21 (2020): 1-12.