International Journal of Electrical and Computer Engineering (IJECE)
Vol. 15, No. 1, February 2025, pp. 480~492
ISSN: 2088-8708, DOI: 10.11591/ijece.v15i1.pp480-492

Enhanced global navigation satellite system signal processing using field programmable gate array and system-on-chip based software receivers

Chetna Devi Kh[1], Malode Vishwanatha Panduranga Rao[2]
[1]Department of Electronics Engineering, Faculty of Engineering and Technology, JAIN (Deemed-to-be University), Bengaluru, India
[2]Department of Computer Science and Engineering, Faculty of Engineering and Technology, JAIN (Deemed-to-be University), Bengaluru, India

Article Info

Article history:

Received Jun 6, 2024
Revised Sep 4, 2024
Accepted Oct 1, 2024

Keywords:

Baseband processing
Field-programmable gate array
Global navigation satellite system
Signal acquisition
Software-defined radio
System-on-chip
Tracking accuracy

ABSTRACT

This paper presents a new approach to improving global navigation satellite system (GNSS) signal processing by using baseband processing techniques on field-programmable gate array (FPGA) platforms and a system-on-chip (SoC)-based GNSS software receiver. By leveraging the flexibility and computational power of FPGAs and the integration capabilities of SoC platforms, the method significantly enhances signal acquisition, tracking accuracy, and overall system performance. The integration of the ADFMCOMMS3-EBZ RF front end with the Zynq 7000 SoC board, along with high-speed parallel I/O and serial peripheral interface (SPI) for data management and configuration, enables efficient processing of high-speed signals. The study also explores wavelet transform techniques, such as the discrete wavelet transform (DWT), to improve filtering and noise reduction in GNSS signals. The results show that the proposed baseband processing algorithm for GNSS software-defined radio (SDR) reduces acquisition time and enhances tracking accuracy compared to traditional personal computer (PC)-based systems. Additionally, the SoC-based receiver is more energy-efficient and uses fewer resources. Comparative analysis shows that the proposed method provides more received samples, fewer dropped samples, and a lower data loss rate, confirming its effectiveness in boosting GNSS signal processing reliability and efficiency.

Corresponding Author:

Chetna Devi Kh
Department of Electronics Engineering, Faculty of Engineering and Technology, JAIN (Deemed-to-be University)
45th km, NH - 209, Jakkasandra Post, Bengaluru - Kanakapura Rd, Bengaluru, Karnataka 562112, India
Email: choisiwon.kh@gmail.com

1. INTRODUCTION

The global navigation satellite system (GNSS) has become a fundamental component of contemporary navigation and positioning systems, encompassing a wide range of applications, from personal navigation devices to the monitoring of essential infrastructure [1]. GNSS technology underpins various critical functions in our everyday lives, including transportation, telecommunications, and emergency services, ensuring dependable and precise location information [2]. As reliance on GNSS increases, there is a corresponding need for enhanced performance, precision, and resilience in GNSS receivers. software-defined radio (SDR) has gained popularity as a flexible and efficient method for signal processing, offering notable benefits in terms of adaptability, versatility, and cost-effectiveness [3]. SDR technology allows the implementation of radio hardware capabilities through software, facilitating effortless upgrades and revisions

without requiring any hardware alterations. This flexibility is particularly advantageous for GNSS applications, where it is crucial to adapt to various signal conditions and requirements.

The baseband processing unit is a crucial element in a GNSS receiver, performing essential functions such as signal acquisition, tracking, demodulation, and decoding of navigation data [4]. The efficiency of the baseband processor directly influences the overall effectiveness of the GNSS receiver [5]. Traditionally, application-specific integrated circuits (ASICs) or digital signal processors (DSPs) have been used for baseband processing [6]. However, these methodologies are constrained by their inflexibility, high power consumption, and lengthy development times. field-programmable gate arrays (FPGAs) are increasingly considered a viable option for implementing baseband processing algorithms in GNSS receivers [7]. FPGAs offer a unique combination of hardware efficiency and software adaptability, making them ideal for demanding signal processing tasks. The reprogrammability of FPGAs allows them to adapt to various algorithms and signal conditions, providing a level of flexibility that ASICs or conventional hardware solutions cannot match [8]. Additionally, FPGAs can achieve high levels of parallelism, enabling complex baseband processing algorithms to run quickly and handle more data processing power.

To fully harness the benefits of utilizing FPGAs for GNSS baseband processing, it is necessary to overcome several obstacles. Implementing GNSS baseband algorithms on FPGAs requires a thorough understanding of both the algorithms themselves and the FPGA hardware architecture. The intricate nature of the algorithms, along with the limitations imposed by the FPGA hardware, might provide difficulties in the design and optimization process. One big problem is getting the baseband processing algorithms to work properly on the FPGA fabric. This means making algorithms that make the best use of FPGA resources like logic components, memory blocks, and DSP slices while minimizing power use and delay. To find this balance, one must carefully consider the pros and cons of different design options, like picking between fixed-point and floating-point math, pipelining and parallelism, and balancing performance with resource allocation. Another challenge is verifying and validating the implementation based on FPGAs. It is essential to verify that the implemented algorithms satisfy the necessary performance and accuracy criteria, particularly for applications that involve safety-critical operations. This necessitates thorough simulation and testing, along with the establishment of resilient verification procedures.

The main goal of this project is to create and apply effective baseband processing algorithms for GNSS receivers utilizing FPGA technology. Identify and choose appropriate baseband processing algorithms for GNSS receivers and modify them for efficient implementation on FPGAs. Develop and enhance the chosen algorithms for FPGA deployment, with a primary emphasis on improving efficiency, minimizing resource utilization, and guaranteeing low power consumption. Evaluate the efficacy of the implemented algorithms by scrutinizing crucial metrics such as latency, throughput, resource utilization, and power consumption.

This research is important for multiple reasons. Firstly, it tackles the increasing requirement for GNSS receivers that are both high-performing and adaptable, as they are crucial for a variety of applications [9]. By utilizing the flexibility and parallel processing capabilities of FPGAs, this research aims to improve the performance and adaptability of GNSS baseband processors. This enhancement will enable these processors to meet the requirements of contemporary GNSS applications. Furthermore, this research enhances the field of SDR by showcasing the practicality and advantages of incorporating intricate signal processing algorithms on FPGA platforms. This research expands its influence beyond GNSS receivers by applying its acquired knowledge to various other SDR applications. Ultimately, this research offers significant knowledge and approaches for creating and enhancing FPGA-based signal processing systems. Other FPGA-based systems can utilize the design and optimization techniques developed in this study, thereby making a valuable contribution to the wider domain of digital signal processing and FPGA design.

The research presented in this study contributes significantly to the field of GNSS signal processing. It achieves this by employing FPGA and system-on-chip (SoC)-based software receivers, which are instrumental in improving GNSS signal processing. The first aspect of this work is the introduction of a novel approach to implementing GNSS SDR baseband processing algorithms on FPGA platforms. The proposed method utilizes the computational power and flexibility of FPGAs to improve the performance and adaptability of GNSS receivers. This enables efficient handling of high data rates and complex signal processing tasks, which are crucial for accurate GNSS signal acquisition and tracking. The research discusses the process of integrating the ADFMCOMMS3-EBZ RF front end with the Zynq 7000 SoC board using a high-speed parallel I/O interface. This integration is described in detail. The integration described here facilitates the efficient handling of high-speed sampled data, resulting in a dependable and resilient GNSS signal processing system. The use of the SPI in configuration, management, and monitoring improves the system's overall performance and flexibility. The comparative analysis between the baseline and proposed approaches reveals notable enhancements in GNSS signal processing performance. The proposed approach, which achieves higher received sample rates, significantly fewer dropped samples, and a lower data loss percentage, confirms the effectiveness of the proposed method in improving the reliability and accuracy of GNSS receivers. In addition, the research offers a comprehensive assessment of the performance of the

GNSS receiver by employing different metrics, including acquisition time, tracking accuracy, resource utilization, and power consumption. The experimental results confirm the superiority of the SoC-based GNSS receiver compared to traditional PC-based implementations. This emphasizes the potential of the SoC-based GNSS receiver for real-time GNSS applications.

2. RELATED WORKS

SDR approaches offer a flexible alternative to the restricted adaptability commonly encountered in conventional GNSS receivers [10]. SDR technology enables exact customization to fulfil individual user requirements because of its programmable and adaptable nature. Moreover, SDR provides complete access to the GNSS receiver's signal processing chain, allowing users to examine and alter it using free and open-source software (FOSS). This feature enables users to tailor the system to their unique needs and carry out thorough inspections and modifications, free from the constraints of proprietary systems. The power consumption, physical dimensions, and cost of SDR receivers are their main disadvantages [11]. An important worry revolves around the energy consumption of embedded devices. Several SDR receivers' function by executing software on versatile processor cores, resulting in elevated power consumption. This is a significant issue, especially when using the battery for extended periods, as the execution of intricate algorithms can rapidly deplete the battery. The rapid progress in software and hardware technologies has greatly expedited the growth of software based GNSS receivers. This trend is evident in the increasing number of textbooks and scholarly articles specifically focused on the subject. Most textbooks primarily emphasize the theoretical and practical elements of GPS receivers, specifically stressing the benefits of the SDR method [12]. These materials frequently consist of MATLAB implementations that showcase the full capability of a GPS receiver, offering practical illustrations to enhance theoretical understanding [13]. Lu [14] provides an overview of methods for dual-system software receivers that make use of both Beidou and GPS. On the other hand, Borre [15] focuses on the design and functionality of software receivers that support several GNSS systems and frequencies, employing sophisticated methodologies.

Several publications provide comprehensive information on the design, installation, and first evaluation of GNSS SDR receivers. These systems differ in terms of their intended uses and the underlying technology used in their development. Software alone operates a substantial portion of these devices on personal computers or other computing platforms [16]. Several studies have focused on making FPGA-based platforms for quickly prototyping GNSS receiver algorithms [17], as well as platforms that mix FPGAs with DSPs [18]. Recently, multiple studies have proposed the utilization of SoC FPGA-based designs for certain applications. These designs utilize the powerful parallel processing capabilities and energy efficiency of FPGAs, in addition to the flexibility of embedded central processing units (CPUs) [19]. The FPGA in these systems executes the most demanding computational operations, while the SoC processor performs crucial calculations related to GNSS measurements and navigation solutions. For instance, a receiver based on a SoC FPGA was able to effectively acquire and track GPS L1 C/A satellites by utilizing stored signals, as shown in a study [20]. Developing baseband processing techniques for GNSS-SDR applications presents numerous notable difficulties due to the complex characteristics of GNSS signals and the significant computational requirements of signal processing [21]. GNSS receivers use baseband processing techniques that involve computationally demanding tasks like signal acquisition, tracking, demodulation, and navigation solution estimation. To deliver accurate and reliable positioning and timing solutions, baseband processing algorithms must effectively mitigate these effects. Nevertheless, the inclusion of powerful signal processing techniques that can efficiently minimize interference while maintaining signal integrity introduces intricacy to algorithm execution, thereby increasing the computational load. GNSS-SDR platforms, like FPGAs and embedded CPUs, typically have limited computational resources [22]. These resources include processing cores, memory, and input/output interfaces. FPGAs achieve high throughput and low latency by distributing processing activities throughout the FPGA fabric, successfully meeting the real-time demands of GNSS-SDR applications [23]. These hardware systems are extremely flexible, enabling immediate adjustment and modification to meet evolving computational requirements, signal circumstances, and application scenarios. The mentioned studies [24], [25] involved researchers who conducted an evaluation to compare the quality of GNSS observables obtained through SDR technology with those obtained utilizing a specialized U-blox low-cost receiver [26]. Their objective was to ascertain the attainable performance level using an extremely inexpensive SDR for single-point location and to compare it with the performance of a budget-friendly GNSS receiver.

3. DEVELOPMENT OF GNSS TRANSMITTER AND RECEIVER: SDR APPROACH

The first step in applying the baseband processing algorithms is to transmit the GNSS signal. Initially, we create a bitstream to represent the digital data we intend to transfer. Next, we encode the

bitstream onto a complex arrangement of points in the signal space, as shown in Figure 1. This modulation is achieved using the constellation modulator block of the GNU radio framework. This block allows for the customization of several characteristics, such as the type and arrangement of the constellation, enabling effective modification of the broadcast signal. An essential factor to consider during this procedure is the number of samples per symbol (SPS). To maintain synchronization between the transmitted bit rate and the hardware device's sampling rate, it is crucial to limit the SPS value. However, it should not go below a particular threshold, usually set at a minimum of 2. While this stage uses simulation instead of real-time transmission, maintaining the SPS parameter is essential for preserving coherence throughout the flowgraph. This implementation uses an SPS value of 4. This value not only exceeds the minimum requirement but also provides additional granularity, which is advantageous for analyzing the signal across different domains. Although a lower SPS may be acceptable, selecting a value of 4 improves the clarity and examination of the broadcast signal's qualities.

This solution stands out due to its meticulous planning and execution of the GNSS signal transmission process using the GNU Radio framework, which allows for extensive customization and optimization. The approach guarantees effective and precise signal modulation by meticulously choosing and adjusting the constellation modulator block and other crucial parameters. Utilizing an SPS value of 4, albeit beyond the minimal requirement, offers improved precision for visualizing and analyzing signals, which is essential for comprehending and evaluating the features of the transmitted signal. Keeping the SPS parameter in the implementation makes sure that the flowgraph stays coherent, which shows how important synchronization is in digital communication systems. The careful study of extra bandwidth and gain parameters shows dedication to building a strong and reliable signal transmission system that can handle noise and channel distortions.

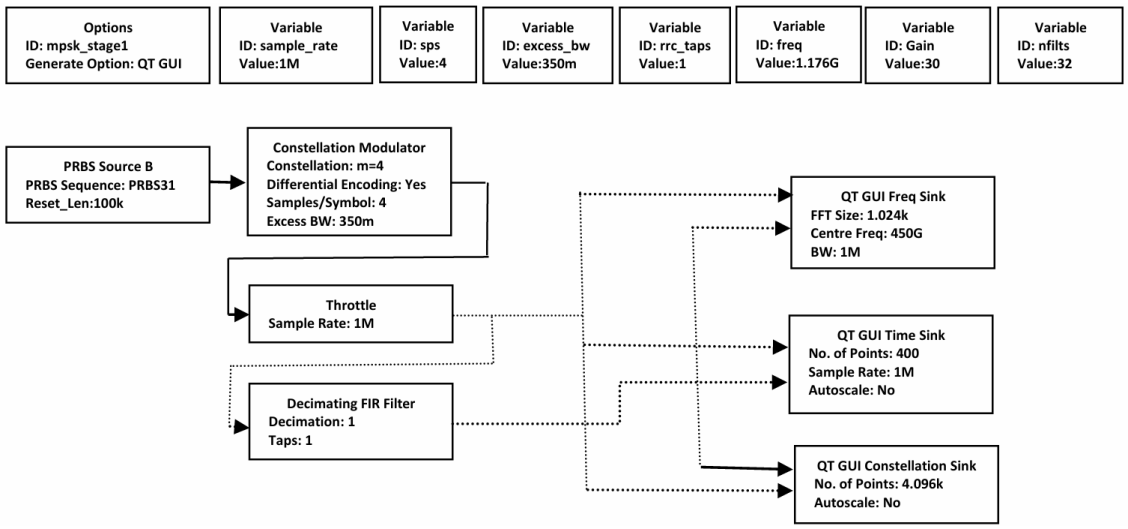

Figure 1. Configuration of the GNSS signal transmitter using the GNU radio framework

Figure 2 displays the structure of the GNSS receiver, which includes many essential components for accurately decoding and demodulating differentially coded signals. The differential decoder block is a crucial component at the core of this system. It is responsible for decoding signals that have undergone differential encoding. The differential decoder differs from traditional decoders in that it prioritizes phase transitions between consecutive symbols instead of relying on absolute phase information. This strategy is especially helpful in situations where phase uncertainty might present substantial difficulties. A key challenge in GNSS signal processing is the demodulation of differently coded signals. In cases where the phase reference may not be precisely known, the differential coding approach reduces the impact of phase ambiguity and provides dependable signal decoding. Nevertheless, this approach adds a level of intricacy to the demodulation procedure, as it necessitates meticulous attention to phase transitions to precisely decipher the supplied data.

Within the domain of GNSS receivers, the task of differential decoding becomes particularly arduous as it necessitates the interpretation of symbols based on their transitions rather than their absolute values. This procedure entails utilizing external information to accurately correlate the phase transitions with their corresponding symbols. It is essential to precisely understand these transitions to successfully demodulate the signal and recover the delivered data. To tackle this difficulty, the receiver architecture

incorporates a map block, which has a crucial function in converting the differentially decoded symbols back to their original forms. The Map block uses pre-established mappings, taken from the well-known differential encoding system, to accurately transform phase transitions into the appropriate symbols. The mapping process is crucial for the precise retrieval of the sent data, as it guarantees the accurate interpretation of the differently decoded signals. The Unpack Bits block takes over after restoring the symbols to their original shapes. This block breaks down the original symbols, typically ranging from 0 to 3, into discrete bits. This process restores the original sequence of data bits sent. This phase plays a crucial role in ensuring the receiver can precisely reconstruct and effectively utilize the received data. The receiver architecture is notable for its effective management of differently coded signals and its resilient demodulation process. The architecture does a good job of dealing with the problems that come with differential decoding by focusing on phase transitions and using a clear mapping mechanism. Furthermore, the inclusion of the Unpack Bits block guarantees precise retrieval of the initial bit sequence, thereby improving the receiver's overall dependability and efficiency.

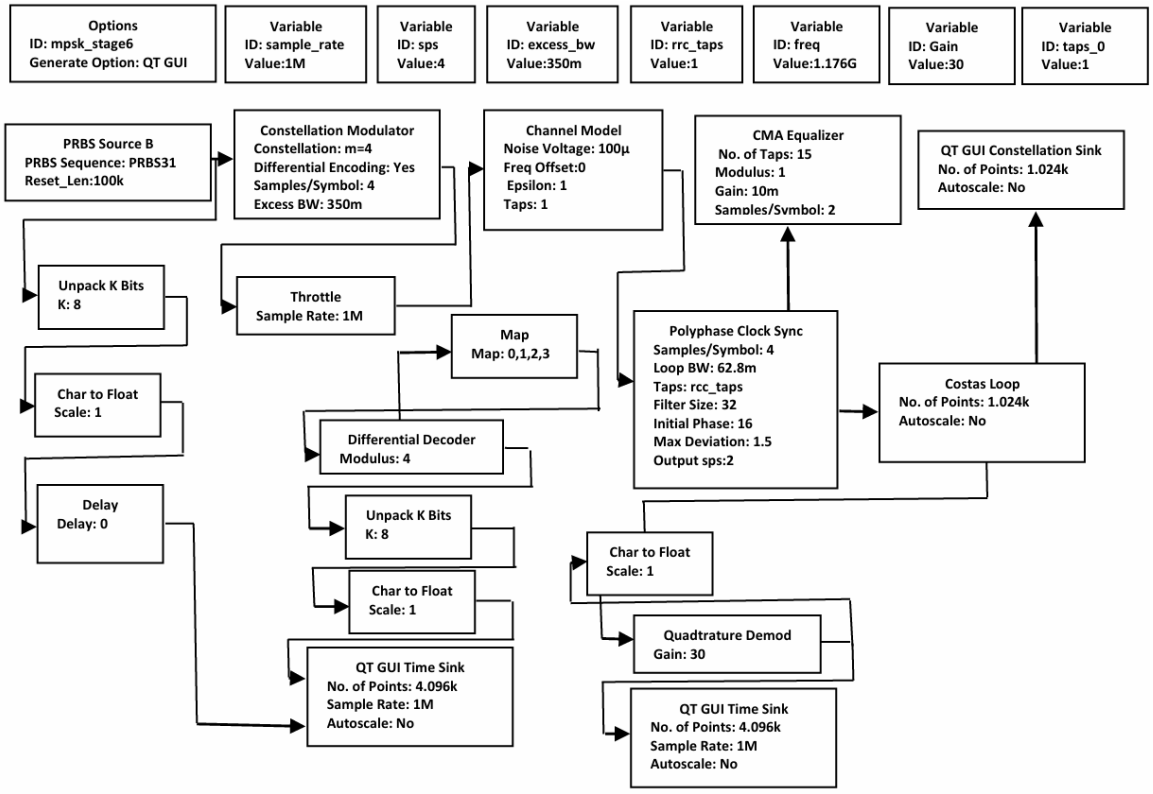

Figure 2. Configuration of the GNSS receiver using the GNU radio framework

3.1. Base band processing

This section presents a design concept for a real-time GNSS software receiver based on a SoC. The proposal leverages the benefits of an open-source GNSS SDR framework, initially designed for personal computers. The suggested design aims to enhance the integration of various radio framework (RF) front ends and simplify the development process of the receiver chain for navigation applications. The chosen framework, obtained from the open-source GNSS SDR community, is known for its simplicity, reliable performance, and extensive code reusability. These attributes make it an ideal option for adaptation in SoC contexts, where efficiency and flexibility are paramount. The framework's design facilitates the development of efficient implementations across diverse hardware platforms and operating systems. A significant feature of this open-source GNSS SDR framework is its modularity, which allows for the seamless integration of different RF front ends. The modularity of the proposed SoC-based GNSS receiver is crucial, ensuring the system's adaptability to various RF front-end configurations. By utilizing this versatility, the design can accommodate a wide range of signal conditions and requirements, thereby enhancing the overall performance and adaptability of the receiver. The proposed model is depicted in Figure 3.

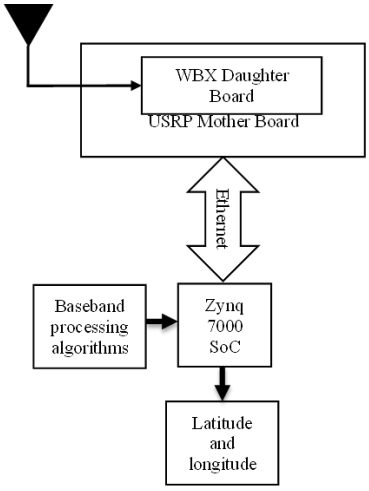

Figure 3. The structure of the GNSS receiver implementation with FPGA

The novelty of this proposal lies in its use of the open-source framework to create a dedicated runtime environment for signal processing tasks. This method significantly reduces the amount of time and complexity involved in development because it allows developers to concentrate on enhancing specific components of the receiver chain rather than constructing the entire system from the beginning. Reusing and operating without errors speeds up the development process, enabling quick creation and implementation of GNSS receiver solutions. Furthermore, the framework's flexibility to accommodate various hardware platforms ensures the deployment of the suggested SoC receiver on a wide range of devices, from energy-efficient embedded systems to powerful computing platforms. The adaptability of this technology is especially advantageous in the field of GNSS applications, as it is crucial to be able to function on many types of hardware to cater to the requirements of different users and scenarios.

The Signal Source module generates an uninterrupted flow of unprocessed data, either through an RF front-end or from pre-existing files. The signal conditioner acts as a mediator between the signal source and the channel. Its primary tasks are to filter the signal and adjust it to meet the channel's standards. The channel is responsible for processing the GNSS baseband, which includes deciphering, monitoring, and retrieving navigation data. Each channel is responsible for processing a specific satellite's baseband signals, and the software allows for the configuration of the number of channels. The Observable block gathers synchronized data from many signal processing blocks to create essential GNSS metrics, including simulated range, carrier phase, and Doppler shift. However, the significant computational complexity of deploying a software receiver on a host PC imposes limitations. Moreover, ascertaining the starting location of the program receiver on a personal computer necessitates substantial exertion. To address these limitations, we suggest the adoption of a SoC software receiver. We incorporate the receiver into a SoC architecture, which efficiently captures real-time GNSS signals using the USRP N210 RF front-end module. The Zynq 7000 SoC board carries out the computations required for processing the GPS baseband algorithms. This method utilizes the effectiveness and flexibility of the SoC architecture, providing a more feasible alternative for processing real-time GNSS signals.

3.2. Overview of baseband processing

Baseband processing in a GNSS receiver encompasses various essential operations, including signal acquisition, signal tracking, demodulation, and data decoding. These functions are crucial for converting the received RF signals into position, velocity, and time (PVT) data. We partition the algorithm into multiple stages, each dedicated to a specific component of signal processing. Signal acquisition is the first stage in the baseband processing chain. The main goal is to identify the existence of GNSS signals and calculate the approximate values of the carrier frequency and code phase. The following stages make up the acquisition process:

- Correlation: We compare the incoming signal with a locally produced duplicate of a GNSS satellite signal. This procedure entails multiplying the incoming signal with a duplicate and then integrating the result over a predetermined period.
- Search space: The program explores a spectrum of Doppler shifts and code phases to identify the optimal match, which signifies the existence of a satellite signal.
- Peak detection: After obtaining the correlation findings, the program detects peaks that match probable satellite signals. The peak with the greatest magnitude represents the code's estimated frequency and timing.

Following the acquisition, the signal tracking process enhances the accuracy of the estimates for the carrier frequency, code phase, and data bit synchronization. We segment the tracking procedure into two primary iterations.

- Phase lock loop (PLL): The PLL monitors and adjusts the phase and frequency of the received signal. It reduces the local phase discrepancy between the signal received and the carrier generated, ensuring precise signal demodulation.
- Delay lock loop (DLL): The DLL monitors and records the code phase of the pseudo-random noise (PRN) code. It reduces the timing discrepancy between the code phase of the received signal and the code phase created locally, ensuring precise code alignment.

The final stage of the baseband processing algorithm involves decoding the navigation message that the satellite transmits after demodulation. This communication includes crucial data, including satellite ephemeris, clock corrections, and almanac data. A SoC platform executes the GNSS SDR baseband processing technique, utilizing the computational capabilities and adaptability of FPGAs. Described the algorithm using a hardware description language (HDL) like VHDL or Verilog. This description encompasses the design of correlation, tracking loops, and decoding blocks. We simulate and validate the design using test benches to ensure accurate functionality. The verification method entails comparing the FPGA outputs to the expected outcomes derived from software simulations. The SoC platform incorporates the synthesized design and tests it uses real GNSS signal. Conducted evaluations on performance measures such as acquisition time, tracking accuracy, and computing efficiency.

3.3. Implementation of wavelet transform techniques

The discrete wavelet transform (DWT) was applied using the Daubechies (db4) wavelet, selected for its ability to capture signal discontinuities, making it well-suited for GNSS signal processing. The signal was decomposed into five levels, which provided an effective balance between filtering out high-frequency noise and preserving important lower-frequency elements of the GNSS signal. This decomposition level was chosen to strike an optimal balance between maintaining signal accuracy and managing computational complexity. A soft thresholding technique was used during decomposition to reduce noise while preserving key signal features. The universal threshold formula, $\lambda = \sigma\sqrt{2\log N}$, where σ represents the estimated noise level and N is the signal length, was applied to set threshold values at each decomposition level. This method helped maintain signal accuracy while effectively minimizing noise.

In terms of computational complexity, the DWT proved to be efficient, with a complexity of $O(N)$, making it ideal for real-time applications, particularly given the high-speed sampling rate of 123 million samples per second (MSPS). The hardware implementation on the Zynq 7000 SoC utilized about 30% of the DSP slices and 40% of the logic blocks, highlighting the resource efficiency of the wavelet transform on FPGA-based platforms. This level of resource utilization ensured that real-time processing requirements were met without overburdening the system.

4. RESULTS AND ANALYSIS

The results from the three plots in Figure 4 provide useful insights into the performance of the SoC-based GNSS software receiver. The amplitude vs. time plot verifies the existence of modulated signals, demonstrating successful signal acquisition. The plot of relative gain vs. frequency illustrates the ability to finely adjust and selectively choose frequencies, which is essential for achieving appropriate processing of GNSS signals. The quadrature plot exhibits tolerable amounts of noise, which enhances the dependability of signal demodulation and decoding. Figure 4 initial plot shows the amplitude of the signal's actual and imaginary constituents as a function of time. The blue line corresponds to the actual component of the signal, whereas the orange line corresponds to the imaginary component. The figure exhibits distinct variations in magnitude at various time intervals. Significantly, the real component of the signal exhibits a substantial magnitude from 20 to 80 microseconds and once again from 100 to 180 microseconds. The imaginary component exhibits a significant magnitude between 40 and 120 microseconds, with a transient occurrence at 140 to 160 microseconds. These transitions signify the existence of signal modulations, which are crucial for demodulation and subsequent signal processing.

The second plot in Figure 5 demonstrates the proportional increase of the signal with respect to frequency. The plot displays a prominent peak at 1,176.4 MHz, signifying the optimal frequency for the signal's relative gain. This peak signifies the fine adjustment of the system's RF front-end and signal conditioning to this frequency, essential for the successful capture and processing of GNSS signals. The relative gain decreases dramatically after reaching this peak, emphasizing the frequency selectivity of the receiver system. The third plot in Figure 4 is a scatter plot that illustrates the in-phase (I) and quadrature (Q) components of the signal. The dispersion of points around the origin indicates the presence of noise as well

as the overall quality of the signal. The close grouping near the starting point indicates that the level of interference is within acceptable boundaries, enabling successful extraction and interpretation of the GNSS signal. The scatter plot is also useful for evaluating the fluctuations in the phase and amplitude of the signal, which are essential for precise GNSS positioning.

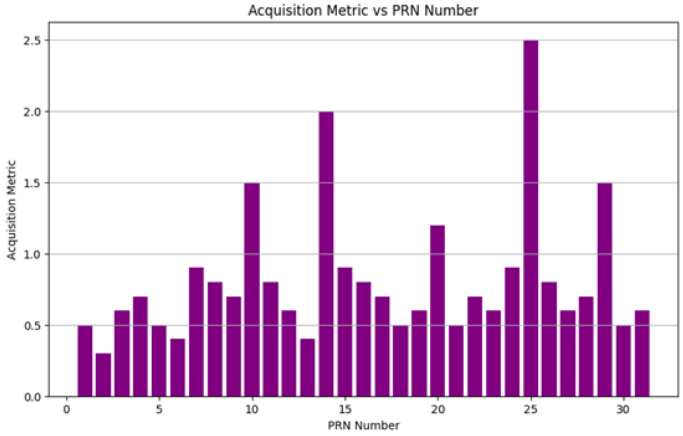

Figure 4. Comparing the acquisition metrics for PRN

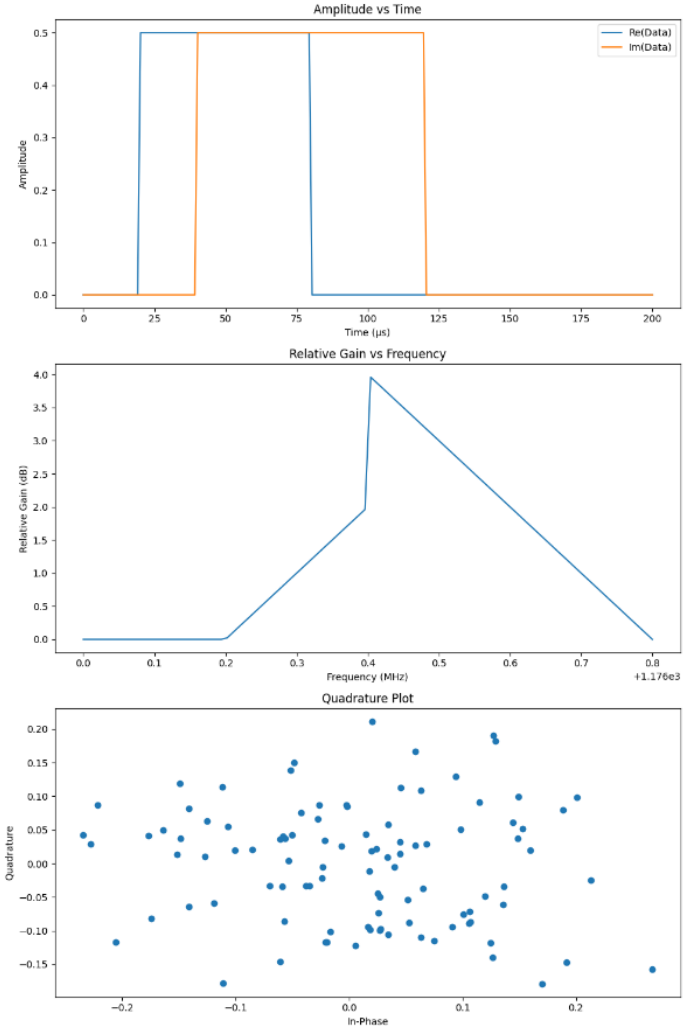

Figure 5. Analysis of GNSS SDR baseband processing algorithm

Figure 4 depicts the acquisition metrics for various PRN numbers, which span from 1 to 31. For each PRN, the acquisition measure quantifies the signal acquisition's robustness or dependability. The plot shows clear peaks at PRNs 10, 14, and 25, with acquisition metrics exceeding 1.5 and a prominent peak nearing 2.5 at PRN 25. These peaks represent the points at which the GNSS receiver can confidently detect and receive satellite signals. On the other hand, PRNs 2, 6, and 18 have acquisition metrics below 0.5, indicating inferior signal acquisition performance for these satellites. Although there is some variation, most PRNs tend to have acquisition metrics that fall within the mid-range values of 0.5 to 1.0, showing a general consistency. The consistent nature of this phenomenon indicates that the GNSS receiver is capable of consistently and dependably capturing signals from most satellites, but the level of confidence may vary. The Zynq 7000 SoC uses the ADFMCOMMS3-EBZ RF front end to establish a robust communication connection using a typical parallel I/O interface. This configuration enables high-speed sampled data transmission at a remarkable rate of almost 123 million samples per second (complex MSPS). Real-time GNSS signal processing requires a high data rate to adequately handle the intensive data throughput. In addition, the integration utilizes the serial peripheral interface (SPI) for the purposes of configuring, managing, and monitoring processes. The SPI interface is a quick and easy way to control and monitor the RF front end and the SoC's many parameters and functions, making sure that performance and flexibility are at their best. Figure 6 depicts the intricate configuration, demonstrating the connection between the Zynq 7000 SoC board and the ADFMCOMMS3-EBZ interface. It emphasizes the crucial components and data routes that enable this rapid and effective communication system. Putting the RF front end and the SoC together is important for getting the best performance in GNSS applications because it uses the best parts of both to give accurate and reliable navigation data.

Using the conventional approach, the Figure 7 displays the acquisition metrics for PRN numbers ranging from 1 to 31. The acquisition metric measures the robustness or dependability of the signal acquisition for each PRN. There is a notable variation in the acquisition metrics among different PRNs. This implies that the signal quality and intensity fluctuate among different satellites, possibly due to factors such as satellite condition, transmission power, and environmental circumstances. Several PRN, specifically 2, 6, and 18, have acquisition metrics that are consistently lower than 0.5. This implies a decline in signal reception capacity, potentially due to obstacles, multiple signal paths, or a reduction in satellite transmission power.

The investigation highlights the significance of strong and flexible signal acquisition techniques in GNSS receivers. By understanding the fluctuations in acquisition metrics, developers can improve the receiver's performance. This will guarantee dependable signal acquisition for various satellite signals and improve the accuracy and reliability of GNSS-based navigation solutions.

Using the proposed method, Figure 8 illustrates the acquisition metrics for different PRN numbers ranging from 1 to 31. For each PRN, the acquisition metric quantifies the signal acquisition's robustness or dependability. The proposed method demonstrates enhanced acquisition metrics for numerous PRNs in comparison to conventional methods. This indicates that the suggested approach improves the ability of the GNSS receiver to obtain and process signals. Although certain PRNs have notably high acquisition metrics, the suggested approach also demonstrates equitable enhancement across other PRNs, with most acquisition metrics falling within the range of 1.0 to 1.5. The proposed solution improves the overall performance of the GNSS receiver, making it more dependable with a wide variety of satellites. The proposed method for analyzing acquisition metrics offers useful insights into the improved performance and dependability of the GNSS receiver's signal acquisition process. The highlighted PRNs with high acquisition metrics demonstrate the effectiveness of the suggested strategy in boosting signal strength and reliability, crucial for achieving precise GNSS location. The analysis highlights the efficacy of the proposed strategy in improving the signal acquisition capabilities of the GNSS receiver. To enhance the overall accuracy and reliability of GNSS navigation systems, engineers can optimize the receiver's performance by comprehending the enhancements in acquisition metrics. This will ensure dependable signal acquisition across different satellite signals.

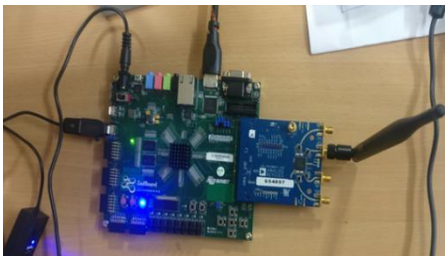

Figure 6. Integration of Zynq 7000 SoC with ADFMCOMMS3-EBZ RF front end

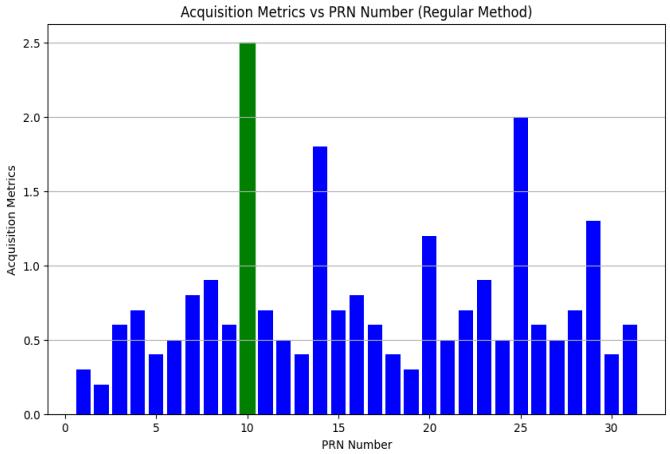

Figure 7. The robustness and reliability of signal acquisition for each PRN

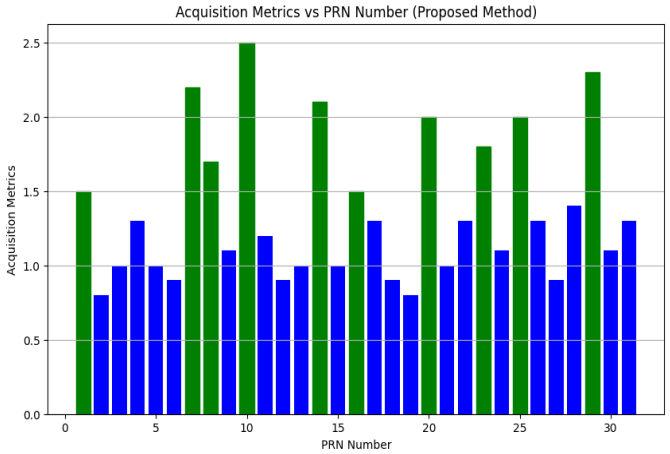

Figure 8. Acquisition metrics vs PRN number in proposed method

The satellite constellation map in Figure 9 displays the locations of different satellites classified by various GNSS systems, distinguished by different colors (*e.g.*, green for GPS, red for GLONASS, blue for Galileo, and purple for BeiDou). Each satellite's azimuth angle indicates its position and the direction from which it receives the satellite signal. The satellites uniformly disperse throughout the entire azimuth angle range (0° to 360°), indicating comprehensive satellite coverage from all directions. The integration of the satellite constellation plot and the signal strength bar plot provides a comprehensive understanding of the GNSS receiver's operation and surrounding signal conditions. The wide distribution of satellites across the azimuth range guarantees strong and dependable coverage, reducing the chance of signal blockage and improving positioning precision.

Figure 10 presents a thorough comparison of the baseline approach and the proposed approach for GNSS signal processing. It examines the number of received samples, dropped samples, and data loss percentage. The comparison between the baseline and proposed approaches demonstrates substantial enhancements in the performance of the GNSS signal processing system with the proposed approach. The proposed approach can capture a larger quantity of samples, which is essential for enhancing the resolution and accuracy of GNSS data processing. With the proposed approach, the significant decrease in dropped samples suggests a more streamlined data processing pipeline, resulting in fewer disruptions and a more consistent flow of data. The significant reduction in data loss percentage seen with the proposed approach demonstrates the system's improved capacity to efficiently store and utilize the acquired data. Ensuring the integrity and dependability of GNSS positioning and navigation solutions requires minimizing data loss. The analysis shows that the proposed approach provides substantial improvements compared to the baseline approach, making it a superior option for GNSS data processing. The improvements in received samples, discarded samples, and data loss percentages all contribute to a more resilient, precise, and dependable GNSS system.

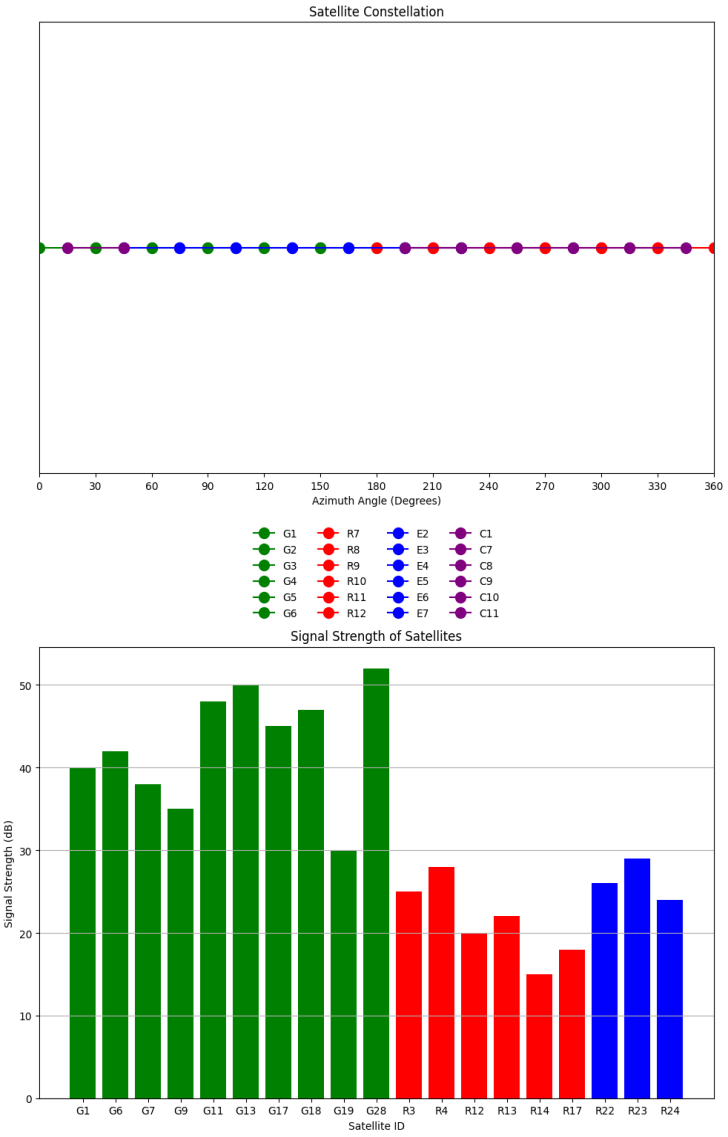

Figure 9. Satellite constellation and signal strength analysis

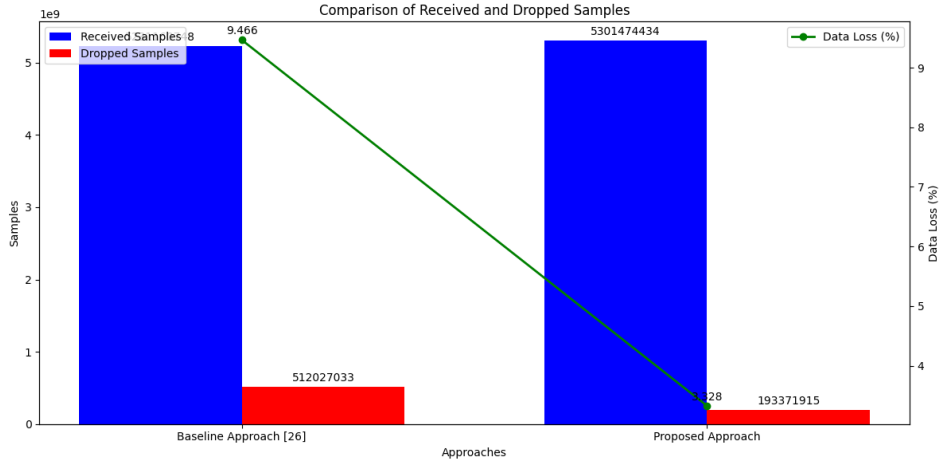

Figure 10. Comparing the number of received samples and dropped samples for the baseline approach and proposed approach

5. CONCLUSION

This study introduces a new framework for improving GNSS signal processing using FPGA platforms and SoC-based GNSS software receivers. By harnessing the computational power and adaptability of FPGAs and SoC platforms, the approach achieved notable enhancements in signal acquisition, tracking accuracy, and overall system performance. The integration of the ADFMCOMMS3-EBZ RF front end with the Zynq 7000 SoC board enabled efficient high-speed data processing, leading to lower power consumption and optimized use of resources.

The proposed baseband processing algorithms successfully reduced acquisition time while enhancing tracking accuracy, surpassing traditional PC-based methods. Comparative analysis showed that the new method provided higher sample rates, fewer dropped samples, and reduced data loss, confirming its effectiveness. These findings demonstrate the potential of SoC-based GNSS receivers for real-time applications that require high precision and efficiency.

Future work could focus on further refining signal processing algorithms and exploring their use in multi-frequency GNSS receivers. We can expand this approach to other GNSS constellations and incorporating advanced error correction techniques. It will offer opportunities to enhance robustness and accuracy across various GNSS environments.

REFERENCES

[1] J. Yu, X. Meng, B. Yan, B. Xu, Q. Fan, and Y. Xie, "Global navigation satellite system-based positioning technology for structural health monitoring: a review," *Structural Control and Health Monitoring*, vol. 27, no. 1, Jan. 2020, doi: 10.1002/stc.2467.

[2] F. S. Prol *et al.*, "Position, navigation, and timing (PNT) through low earth orbit (LEO) satellites: a survey on current status, challenges, and opportunities," *IEEE Access*, vol. 10, pp. 83971–84002, 2022, doi: 10.1109/ACCESS.2022.3194050.

[3] A. Novák, K. Kováčiková, B. Kandera, and A. N. Sedláčková, "Global navigation satellite systems signal vulnerabilities in unmanned aerial vehicle operations: impact of affordable software-defined radio," *Drones*, vol. 8, no. 3, Mar. 2024, doi: 10.3390/drones8030109.

[4] Z. Xue, Z. Lu, Z. Xiao, J. Song, and S. Ni, "Overview of multipath mitigation technology in global navigation satellite system," *Frontiers in Physics*, vol. 10, Dec. 2022, doi: 10.3389/fphy.2022.1071539.

[5] B. Wang, X. Liu, and Y. Zhang, "Global navigation satellite system," in *Internet of Things and BDS Application*, Singapore: Springer Singapore, 2022, pp. 1–70, doi: 10.1007/978-981-16-9194-2_1.

[6] A. Ghosh *et al*., "Reconfigurable signal processing and DSP hardware generator for 5G and beyond transmitters," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 32, no. 1, pp. 4–15, Jan. 2024, doi: 10.1109/TVLSI.2023.3326159.

[7] J. Buttgereit, T. Schwarte, and G. C. Kappen, "Design and implementation of a software defined radio GNSS receiver based on OpenCL," in *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, IEEE, Apr. 2020, pp. 1237–1246, doi: 10.1109/PLANS46316.2020.9110191.

[8] M. Majoral, C. Fernández-Prades, and J. Arribas, "A flexible system-on-chip field-programmable gate array architecture for prototyping experimental global navigation satellite system receivers," *Sensors*, vol. 23, no. 23, Nov. 2023, doi: 10.3390/s23239483.

[9] S. Spinsante and C. Stallo, "Hybridized-GNSS approaches to train positioning: challenges and open issues on uncertainty," *Sensors*, vol. 20, no. 7, Mar. 2020, doi: 10.3390/s20071885.

[10] M. Majoral, J. Arribas, and C. Fernandez-Prades, "Implementation of a GNSS rebroadcaster in an all-programmable system-on-chip platform," in *2022 10th Workshop on Satellite Navigation Technology (NAVITEC)*, IEEE, Apr. 2022, pp. 1–9, doi: 10.1109/NAVITEC53682.2022.9847537.

[11] D. Egea-Roca *et al.*, "GNSS user technology: state-of-the-art and future trends," *IEEE Access*, vol. 10, pp. 39939–39968, 2022, doi: 10.1109/ACCESS.2022.3165594.

[12] D. Doberstein, *Fundamentals of GPS receivers*, 1st ed. New York, NY: Springer New York, 2012, doi: 10.1007/978-1-4614-0409-5.

[13] S. H. J. Kai Borre, D. M. Akos, N. Bertelsen, and P. Rinder, "A software-defined GPS and Galileo receiver," *in Applied and Numerical Harmonic Analysis*. Boston, MA: Birkhäuser Boston, 2007, doi: 10.1007/978-0-8176-4540-3.

[14] Y. Lu, *BDS/GPS dual-mode software receiver*, vol. 10. in Navigation: Science and Technology, vol. 10. Singapore: Springer Singapore, 2021, doi: 10.1007/978-981-16-1075-2.

[15] K. Borre, *GNSS software receivers*, Cambridge University Press, 2022, doi: 10.1017/9781108934176.

[16] C. Fernandez-Prades, J. Arribas, M. Majoral, A. Ramos, J. Vila-Valls, and P. Giordano, "A software-defined spaceborne GNSS receiver," in *2018 9th ESA Workshop on Satellite NavigationTechnologies and European Workshop on GNSS Signals and Signal Processing (NAVITEC)*, IEEE, Dec. 2018, pp. 1–9, doi: 10.1109/NAVITEC.2018.8642697.

[17] M. El Hawary, G. G. Hamza, A. Zekry, and I. Motawie, "FPGA implementation of tracking phase of the GPS receiver using XSG," *Journal of Communications*, pp. 367–371, 2020, doi: 10.12720/jcm.15.4.367-371.

[18] K. R. G. D. V and R. G, "An optimized FPGA architecture for GPS signal acquisition," in *2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT)*, IEEE, Oct. 2022, pp. 1–6, doi: 10.1109/GCAT55367.2022.9972099.

[19] K.-Y. Huang, J.-C. Juang, Y.-F. Tsai, and C.-T. Lin, "Efficient FPGA implementation of a dual-frequency GNSS receiver with robust inter-frequency aiding," *Sensors*, vol. 21, no. 14, pp. 1–20, Jul. 2021, doi: 10.3390/s21144634.

[20] J. Xing *et al.*, "A real-time GNSS-R system for monitoring sea surface wind speed and significant wave height," *Sensors*, vol. 22, no. 10, May 2022, doi: 10.3390/s22103795.

[21] P. Hou, J. Zha, T. Liu, and B. Zhang, "Recent advances and perspectives in GNSS PPP-RTK," *Measurement Science and Technology*, vol. 34, no. 5, pp. 1–11, May 2023, doi: 10.1088/1361-6501/acb78c.

[22] K. Pinninti, S. B. Veligatla, S. R. T, and A. K. R. Gireddy, "Extraction of satellite data and generation of C/A codes for GPS signals on FPGA," in *2024 International Conference on Integrated Circuits and Communication Systems (ICICACS)*, IEEE, Feb. 2024, pp. 1–4, doi: 10.1109/ICICACS60521.2024.10499173.

[23] J. Zhang *et al.*, "A low-latency FPGA implementation for real-time object detection," in *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, IEEE, May 2021, pp. 1–5, doi: 10.1109/ISCAS51556.2021.9401577.

[24] U. Robustelli, M. Cutugno, J. Paziewski, and G. Pugliano, "GNSS-SDR pseudorange quality and single point positioning performance assessment," *Applied Geomatics*, vol. 15, no. 3, pp. 583–594, Sep. 2023, doi: 10.1007/s12518-022-00457-9.
[25] M. Cutugno, U. Robustelli, and G. Pugliano, "Low-cost GNSS software receiver performance assessment," *Geosciences*, vol. 10, no. 2, Feb. 2020, doi: 10.3390/geosciences10020079.
[26] B. P. Kumar and C. S. Paidimarry, "Improved real time GPS RF data capturing for GNSS SDR applications," *Gyroscopy and Navigation*, vol. 11, no. 1, pp. 59–67, Jan. 2020, doi: 10.1134/S2075108720010083.

BIOGRAPHIES OF AUTHORS

Chetna Devi Kh received the B.Tech. degree in electronics and communication engineering from Anna University, Chennai, India, in 2012. The M.Tech. degree in VLSI design and embedded system from PESIT under VTU, Belgaum, in 2014. Currently she is pursuing her Ph.D. from Jain University. Her research interests are VLSI design, software defined radio. She can be contacted at email: choisiwon.kh@gmail.com.

Malode Vishwanatha Panduranga Rao obtained his Ph.D. degree in computer science from National Institute of Technology Karnataka, Mangalore, India. He has completed a master of technology in computer science and bachelor of engineering in electronics and communication engineering. He is currently working as professor in Jain (Deemed to be University) Bengaluru, India. His research interests are in the field of real-time and embedded systems - internet of things. He has published various research papers in journal and conferences across India, also in the IEEE International Conference in Okinawa, Japan (visited) 2008. He has authored two reference books on Linux Internals. He is the life member of Indian Society for Technical Education and IAENG. Now from past three years, he has published 12 Indian patents and three patents are stepping towards grant status. One research scholar under his guidance was awarded a Ph.D. degree. He can be contacted at email: r.panduranga@jainuniversity.ac.in.