Bulletin of Electrical Engineering and Informatics
Vol. 13, No. 5, October 2024, pp. 3233~3242
ISSN: 2302-9285, DOI: 10.11591/eei.v13i5.7074

Enhancing radar applications: FPGA-driven phase estimation with floating point arithmetic

Ponduri Sivaprasad[1,2], Anandi Venkataraman[2,3], P Satyanarayana Murty[2,4]
[1]Department of Electronic and Computer Engineering, RVR&JC College of Engineering, Guntur, India
[2]Visvesvaraya Technological University, Belagavi, Karnataka, India
[3]Department of Electronic and Computer Engineering, Ramaiah Institute of Technology, MSRIT Post, Bangalore, India
[4]Gonna Institute of Information Technology and Sciences, Visakhapatnam, India

Article Info

Article history:

Received Jun 26, 2023
Revised Dec 20, 2023
Accepted Feb 29, 2024

Keywords:

Field programmable gate array
Floating point unit
Milli-degree phase estimation
Radar applications
Very-large-scale integration

ABSTRACT

This article introduces a paradigm shift in radar technology with field programmable gate array (FPGA)-driven Phase estimation using floating point arithmetic (FPA). Leveraging FPGA's parallel processing and the precision of FPA, this work promises enhanced accuracy and efficiency. The proposed system's key performance metrics include the following: number of slices: 20,941, number of look-up tables (LUTs): 22,371, number of digital signal processing (DSP) blocks: 2, delay: 112.9 ns, and power consumption: 7.2 mw. A comparative analysis showcases advantages in area utilization, LUT, and DSP blocks despite a trade-off with delay. The presented methodology and results demonstrate the feasibility of real-time phase estimation at GHz rates, positioning this approach as transformative for next-gen radar systems.

Corresponding Author:

Ponduri Sivaprasad
Visvesvaraya Technological University
Belagavi, Karnataka-590018, India
Email: psivaprasad@rvrjc.ac.in

1. INTRODUCTION

The merger of cutting-edge hardware and complex algorithms has given rise to an innovative field programmable gate array (FPGA)-driven phase estimation with floating point arithmetic (FPA) in the ever-evolving field of radar technology, where precision is vital. This dynamic combination can potentially take radar applications to new heights by providing a quantum leap in accuracy and efficiency. As we explore deeper into the complexities of this transformative synergy, it becomes clear that a new era of radar capabilities has arrived [1].

The FPGA, a reconfigurable semiconductor technology that unlocks Pandora's box of possibilities for radar applications, is central to this revolution. Unlike traditional processors, FPGAs have parallel processing capabilities that are naturally parallel with radar signal processing. This parallelism results in extremely high processing speeds, allowing for real-time data analysis and decision-making. Before delving into the specifics, it is critical to understand radar technology. Radar, also known as radio detection and ranging, is a device that detects, locates, and tracks things ranging from aircraft and ships to weather patterns [2]–[5]. The continuous need for improved radar systems has been driven by various factors, including: i) precision: with military, aviation, and maritime applications, radar systems must be exact to ensure the safety and security of people and assets; ii) bandwidth: as the world becomes more interconnected, radar systems require wider bandwidths to handle many signals and information; iii) adaptability: modern radar systems must adapt swiftly to different scenarios and changing environments, making flexibility essential;

and iv) performance: high-performance radar systems are essential for achieving optimal defense, meteorology, and space exploration results.

FPA adds another degree of sophistication to the FPGA-driven phase estimation, ushering in previously impossible precision. Unlike fixed-point arithmetic, which has a fixed number of bits for integer and fractional components, FPA has a variable range of values. This adaptability greatly expands numerical data representation, lowering quantization mistakes and improving phase estimation accuracy. Because of its versatility and reconfigurability, FPGAs have recently become popular. This literature review will examine current research on memory core development for next-generation radar applications employing FPGA.

Huang *et al.* [6] suggested improving time-efficiency of variational specific differential phase estimation. Choi *et al.* [7] suggested an FPGA-based range-Doppler algorithm implementation for real-time synthetic aperture radar (SAR) imaging. Chan *et al.* [8] offer a hardware version of a SAR processor based on FPGAs. Time-series clustering methodology for estimating atmospheric phase screen in ground-based In SAR data [9].

Kim [10] provide an overview of the recent trends in memory design for various emerging applications, including internet of things (IoTs), wearable devices, and neural networks. It highlights the challenges conventional memory faces, such as static random-access memory (SRAM), dynamic random-access memory (DRAM), and flash memory in meeting the requirements of these applications. NXP has developed the S32R processor, which is a 32-bit power architecture-based microcontroller for automotive and industrial radar applications. It offers an increased level of integration available to designers of next-generation automotive radar modules [11]. Mukherjee [12] discusses important new radar technologies such as multiple inputs, multiple output (MIMO) systems, digital beam forming (DBF) techniques, active electronically steered array (AESA) radar, millimeter-wave radar, passive coherent location radar (PCLR) systems, semiconductor power amplifiers (PA), intelligent signal coding and radar DSP. Leonardo [13] discusses the most promising upcoming developments in the field of radar technology, including the impact of digitalization, the latest-generation multifunctional antennas, radar applications to support sustainable development, and, in the longer term, quantum radars.

FPGAs possess distinctive features like adaptability, energy efficiency for specific applications, and a remarkable data processing speed. These attributes enable users to execute a wide range of direct-control functions in building digital systems, including digital SAR processors. NASA and the USAF have recognized the critical need to enhance SAR system performance by advancing onboard processing technologies [14]. In 2012, Lesnik and colleagues at the Military University of Technology in Poland developed and put into operation a real-time SAR image processor using readily available commercial off-the-shelf (COTS) hardware, specifically the Xilinx FPGA Virtex-5 SX95T [15]. In 2016, NASA introduced UAVSAR, an airborne SAR testbed with a unique onboard processor (OBP) design. This OBP featured a hybrid architecture that utilized FPGAs for rapid and repetitive tasks and a microprocessor with a floating-point coprocessor for less frequent and irregular computations [16]. Yang *et al.* [17] propose a partial fixed-point processing scheme, using fixed-point for the computationally intensive fast fourier transform (FFT) operation, and demonstrate the system's fidelity and accuracy compared to conventional ground-based software processors, achieving impressive results in terms of processing speed and power consumption for SAR imaging tasks. Liu *et al.* [18] describes the development of a MiniSAR signal processing system using FPGA technology, implementing the polar format algorithm (PFA) and advanced techniques for high-precision and high-speed SAR signal processing.

A semi-parallel iterative decimal multiplier is introduced, utilizing BCD-8421 encoding and a novel partial product reduction technique to achieve faster multiplication, with the design validated on an FPGA, outperforming other multipliers discussed in the research [19]. Another notable development in [20] is using FPGAs with embedded processing units, such as ARM cores. These embedded processors can offload some of the memory-related tasks and enable more efficient memory management, reducing the FPGA's resource and power consumption.

For radar data storage and retrieval, data compression techniques have been investigated as a way to maximize memory utilization and minimize power consumption. For FPGA-based radar systems, algorithms such as run-length encoding (RLE) and Huffman coding have been modified to enable effective data transmission and storage [21]. Research is still being done on creating ultra-low-power FPGAs designed especially for battery-operated and portable radar applications [22]. These FPGAs enable longer mission durations with less energy usage by combining low-leakage power FPGA fabric with power-efficient memory cores.

The motivation behind developing a high-performance phase estimation technique based on FPA is to overcome the limitations of traditional fixed-point phase detectors. Fixed-point arithmetic can introduce quantization errors, especially when dealing with high frequency signals or nuanced phase differences. These

errors can lead to reduced accuracy and degraded system performance. This article aims to achieve greater precision in phase estimation by utilizing FPA, leading to more reliable results in various applications.

In this article, we delve into developing a high-performance phase estimation technique that utilizes FPA to reduce the phase error path in standard phase detection adders. The ultimate goal is to achieve accurate and precise phase estimation with low computational overhead, which is crucial for a wide range of applications, including next-generation radar applications, communication systems, radar, and signal processing.

The main objective of this article is: i) to ensure the compatibility of phase estimation block over a wide range of high-rate (several GHz) space-borne applications; ii) to create a programmable-rate all digital phase locked loop (ADPLL) that uses first in first out (FIFO) and finite-state machine (FSM)-driven RAM to process high-rate samples; and iii) to develop IEEE 754 floating point arithmetic with optimal integer to floating point conversion for phase estimation. The structure of this paper is as follows: section 2 provides the methodology in detail. Section 3 covers the hardware implementation and experimental validation results, and the paper concludes in section 4.

2. METHOD

It begins with representing the input signals and the phase reference using floating-point numbers. This representation allows for a wider dynamic range and finer precision, mitigating the quantization errors associated with fixed-point arithmetic. Custom-designed floating-point units within the FPGA are used to maintain computational efficiency. Once the input signals are represented in floating-point format, proceed to calculate the phase difference between them. This is achieved by subtracting the phase of the reference signal from the phase of the input signal. The result is a floating-point value that represents the phase difference.

Novel error correction techniques are implemented to reduce phase errors further. These techniques involve pre-processing the input signals to minimize phase errors introduced by the hardware and to enhance the accuracy of the phase difference calculation. With the algorithm developed and the error correction mechanisms in place, move on to the FPGA hardware synthesis phase. Implement the entire phase estimation system on an FPGA platform, which enables real-time phase estimation at GHz rates. This ensures that the proposed method is suitable for high-frequency applications. A set number of significant digits denotes a floating-point number and is proportionally adjusted using an exponent in a predetermined base. Xilinx tools endorse three formats: half (16-bit), single (32-bit), and double (64-bit) [4]. To illustrate, a single-format number is expressed as shown in Figure 1.

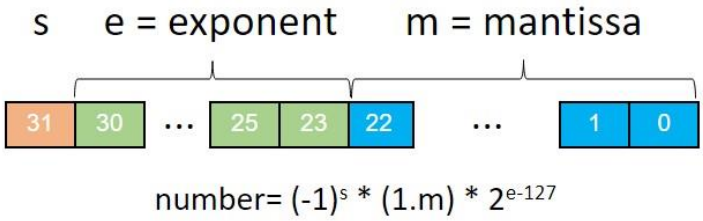


Figure 1. FPA [4]

Dealing with floating-point-based algorithms can be more intricate compared to fixed-point, particularly when employing hardware description language (HDL) such as VHSIC hardware description language (VHDL) or verilog. Thankfully, several Xilinx tools, including the vivado floating-point IP and other high-level synthesis (HLS) tools, significantly simplify the development of algorithms based on floating-point precision. These tools introduce an abstraction layer, eliminating the necessity for users to navigate the intricacies of data representations down to their binary levels. This convenience streamlines the development process and allows users to focus more on the algorithmic aspects rather than the low-level data manipulation.

2.1. Block-based phase detection algorithm

In this case, phase detection uses a block-based strategy in which sample blocks are generated from input signals. M number of successive samples are included in every sample block. In this instance, the M-value is set to 2,400 and can accommodate many input signal cycles for a specified sampling frequency, as shown in Figure 2.

2.2. Reconfigurable clock divider

Furthermore, every unit in the suggested phase estimation system operates at it is rate, and the rate of delay buffer chain depth transition is controlled following the incoming clock rate. All forms of sampling rates for input samples are accepted, and the effectiveness of the real-time data propagation process is verified using suitable clock-down sampling as well as synchronization events. It has also been demonstrated that the most excellent overall performance for next-generation networks is in the multiple GHz range. Depending on the expected samples, the incoming clock rate and the maximum operating speed of the buffer state transition can be calculated. Adaptive optimization must utilize variable delay lines for the various input clock rates.

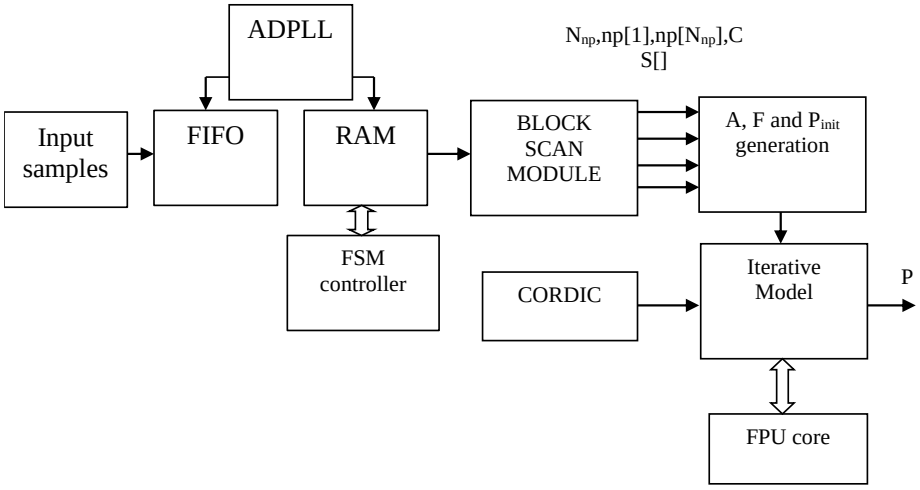


Figure 2. A block-based phase detection system

2.3. All digital phase-locked loop

Analog circuitry acts as voltage-controlled oscillators to divide input frequency. The adaptive line selection based on a fully reconfigurable all-digital frequency divider is suggested to address the prerequisites of dynamic range requirement and limit speed in phase detection systems. Inverters, as well as gate chains, make up the input sample rate that is driven by the rate-controller outputs delay line. To achieve the clock skew design requirements, chains are linked depending on the unit delay for every logic cell. Consequently, a sophisticated delay control system is used. Phase match techniques are employed based on reconfigurable delay lines and highly reconfigurable clock dividers.

- A synchronous phase estimation system with FIFO architecture is adopted to overcome the drawback of asynchronous in a subdivided low-rate phase detection model based on block.
- With this paradigm, multi-rate clock domains can use a single source clock.
- A reconfigurable clock divider can separate a source clock into many number integers.

2.4. Issues in fixed point phase estimation models

2.4.1. Fixed-point precision loss

Suitable bit widths for the representation of optimal signal measurements in fixed point models must adhere to specific criteria for a particular level of accuracy. The precision setback in fixed point outcomes depends on how these intermediate values communicate with the system and the bit width size employed. Any repeated calculations, such as phase detection technique based on the block, will result in accumulations of this error across the entire digital circuit that are wide in range.

2.4.2. Precision

It addresses the number of bits employed to indicate fractional decimal digits. This deals with mathematical accuracy; in the digital realm, it is always finite and uses only a few possible binary digits to enable accurate representation. Whenever addressing the floating-point accuracy, consider the number of mantissa bits. Fixed point models are inappropriate for phase estimation due to the milli-degree level precision requirements.

2.4.3. Accuracy

The degree to which a real phase value and its predicted value in a hardware implementation are similar is referred to as accuracy. Only rounding operations, which are directly relevant to precision, cause a precision loss in any digital system that uses fixed-point arithmetic.

2.4.4. Quantization error

The exact difference between the original value as well as the quantized value is called the quantization error. Digital fractional bits are rounded off with a finite precision loss to integer values. The total performance of any application in real-time depends on the level of accuracy used for the arithmetic computations and the cost-effective implementation of hardware. Usually, FPA is most commonly used in several applications that require phase detection with less complexity in operation. However, for a space-based system, the functional operation has to be maintained accurately to categorize the information collected. In comparison with other existing methods, the Phase shift method is an important approach in space-borne systems for assuring the performance of the system. But this frequently causes many potential challenges. These problems, especially the size of the word, majorly affect the implementation in digital form and cause problems in the computations without errors and the need for hardware with reduced cost.

To address this issue, FPA with high preciseness has to be used. In addition, problems arise due to the non-synchronization of clocks between the data sampled (high rate-in GHz) and the system clock, which causes challenges in implementing FPGA hardware. The presented framework for detecting the phase. It estimates the phases using the IEEE-754 standard, which is of double precision in computing arithmetic operations, and the mille degree was computed by maintaining the discrimination.

3. RESULTS AND DISCUSSION

With the help of a verilog HDL script and a Xilinx FPGA synthesizer, the model has been implemented using a Zybo-7000 FPGA board device from the Vivado Design suite family. Functional verification is typically performed utilizing a thorough test bench enabled by a bit stream. Using the results of simulations, it is well demonstrated that clock rate limitations have intrinsic and variable rates.

3.1. Hardware synthesis results

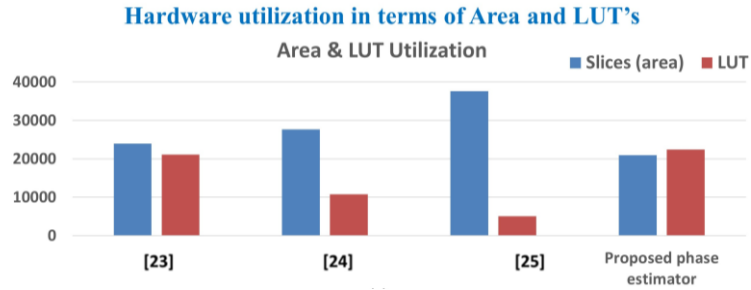
The outcomes obtained from pre-computed data sampling are saved in memory and further utilized as the fundamental building blocks for the phase identification technique. This design process can be easily enhanced throughout hardware synthesis via onboard block RAMs or LUTs available in all FPGA devices. Furthermore, the FIFO-facilitated sub-block divided parallel process is used to validate the variable rate pattern matching system. Also, the functional simulation with additional test vectors has been used to demonstrate the system's effectiveness. Hardware synthesis is used to validate the efficacy of an input sampling based on all digital PLL synthesizers, shown further to be in the multiple GHz range to enable next-generation networks. Additionally, the number of samples employed and the precision rate have a significant impact on computational accuracy as well as on overall power savings. The input sample rates are observed, and configurations are made in response to this technique. Starting with the clock division and moving down to the bottom phase match, every cycle has been adjusted via delay buffers, and certain delay bounds are indicated. Table 1 provides a comparative analysis of the performance of our proposed block-based phase estimator driven by FPA with that of existing literature in terms of area, LUT, No. of DSP block, delay, and power.

Table 1. Comparison of proposed FPA-driven block-based phase estimator's performance with existing literature

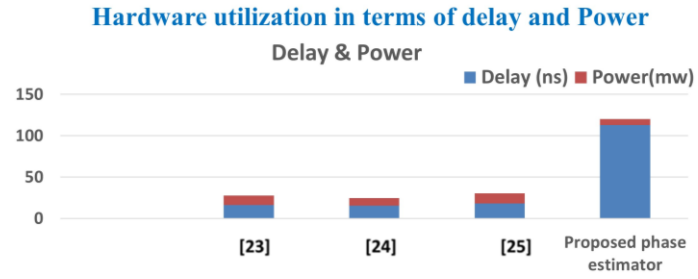
Algorithms	Slices (area)	LUT	DSP blocks	Delay (ns)	Area* delay	Power (mw)
[23]	23,901	21,092	20	16.1	3,84,806.1	11.4
[24]	27,623	10,712	--	15.3	4,22,631.9	9.3
[25]	37,612	5,011	51	18.2	6,84,538.4	12.12
This work	20,941	22,371	2	112.9	23,64,238.9	7.2

Comparing the proposed FPA-driven block-based phase estimator with existing literature, as shown in Figure 3, it is evident that the new approach exhibits notable advantages in terms of area and LUT utilization as shown in Figure 3(a), and DSP blocks. However, it is essential to consider the trade-off with delay, as the proposed method shows a higher delay compared to some existing algorithms. The product of area and delay (Area*Delay) as shown in Figure 3(b) is also significantly influenced by this trade-off. Power

consumption, on the other hand, is relatively lower in the proposed work, indicating potential efficiency in power management.



(a)



(b)

Figure 3. Hardware utilization: (a) area and LUT utilization and (b) dealy and power

The implementation of the parallel phase detection process for continuous data computations has been presented on the other hand. This model's adaptive features enable it to overcome throughput-related issues more precisely and reliably detect the phase of any length input waveform patterns. The performance of hardware is evaluated based on the application of logic elements and logical registers. Further, the analysis of throughput rate demonstrates the advantages of delay optimization by bit-wise state transition over highly optimized bit-based.

3.2. Variable clock rate with synchronous module

As explained previously, the phase detection algorithm based on the block has been implemented using a FIFO-driven FSM as it is designed as distinct identical networks accompanied by ADC calculations. The concept of utilizing special hardware accelerator modules is implemented to address the issues related to clock mismatch. As illustrated in Figure 2, a FIFO enabled by a variable rate clock and its synchronizations with subsequent processing blocks with ADPLL system components are validated. However, once all of the pre-processing blocks have been synchronized, the input phase must match the clock's phase during clock division via ADPLL. Regardless of the input sample rate, the asynchronous payload matches with inconsistent global and localized clocks, resulting in misdetection issues.

3.2.1. Variable clock rate with synchronous module

From the above discussion, the block-based algorithm for detecting the phase is implemented with FIFO-FSM as it is an identical separate network built along with the computations for ADC. In this, the issues of clock mismatching are minimized by the application of dedicated accelerator hardware units. Also, the synchronization of FIFO of variable frequency with ADPLL processing successive block components is verified and shown in Figure 4. After synchronization of all blocks of preprocessing, it is required to utilize the clock of a suitable phase for dividing the clock with ADPLL to match with the phase of the input signal. Misdetection issues arise when the matching of asynchronous payloads with uneven local and globalized watches is done without the concern of input sampling rate.

Again, for the validation of varying delay lines in the process of matching and to perform verification of the effect of asynchronous mismatch in the phase that will happen at any level of accumulating data by using FIFO and generating a clock signal of varying rate is simulated in several dynamics of frequencies.

3.2.2. Phase mismatching in the PFD module

Based on the experiments conducted many times, the PFD is not the most time-consuming operation, as it consists of basic modulo operations that are needed for detecting the phase in synchronous arrays. In the process of detecting the phase, the outcome of the steps used for processing is accelerated, and related execution leads to wrong detection because of the lack of appropriate data collection. UP/DOWN signals are generated by the PFD block and are used in counter-control units. To enhance reconfigurability and scale, every counter unit is adopted for each ADPLL based on the range of the dynamic frequency and the actual amount of locking time, which is shown in Figure 5.

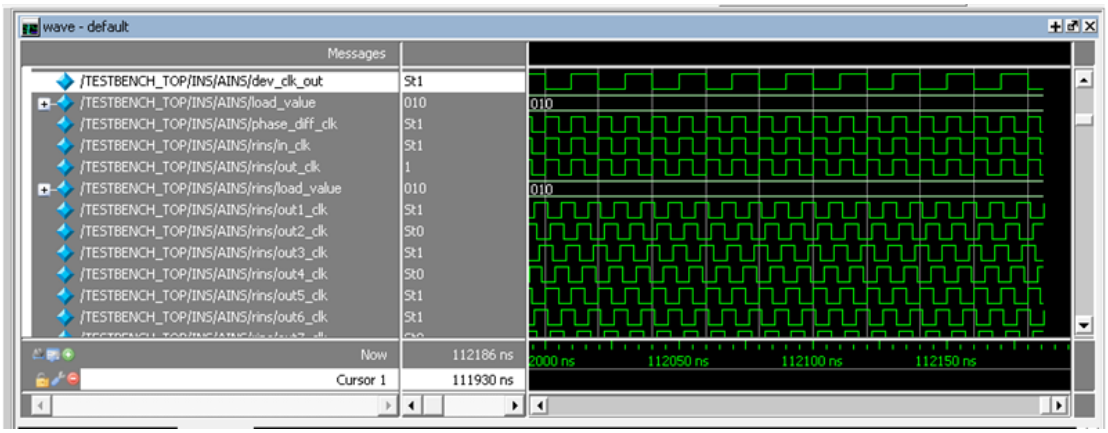


Figure 4. ADPLL down-rated clock

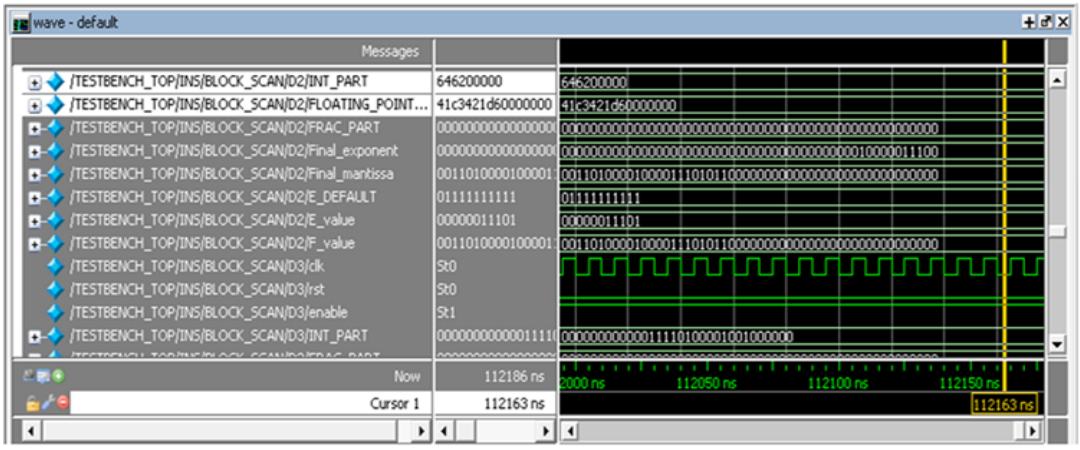


Figure 5. Integer to IEEE 754 double precision floating point conversion output

3.2.3. Floating point division reconstructed simulated signal

BitWise FSM state transitions fail. Even a one-bit mismatch during string matching causes FSM state transitions to fail, and subsequent payloads are skipped over without undergoing any further FSM state transitions or string-matching processes, as shown in Figure 6.

3.2.4. Delay optimization

Generally, the development of the series of stages causes performance degradations in sequential data transfer systems. Enhanced system efficiency is achieved by the bit-paralyzed FIFO as well as the computational data model. In contrast, the number of sequentially dependent FSN state transition operations is decreased, thereby reducing the overall system critical path. With the help of the timing analyzer tool, the maximum operating frequency report can be evaluated. Furthermore, the drawbacks associated with global clock integration are overcome by introducing a phase-synchronized pattern-matching mechanism and a sample rate-driven validation check that is included without creating serious misdetection issues. The

synthesized outcomes showed that the suggested phase detection system achieves enhanced results regardless of the logic register's complexity and Logic element utilization levels. The path delay analysis demonstrates that the proposed core provides significantly better results. The suggested phase detection unit substantially reduces the propagation delay by employing the parallel computation concept. Furthermore, the propagation delay that occurs upon data transition is minimized in addition to the computational delay that results from FPU arithmetic, as shown in Figure 7.

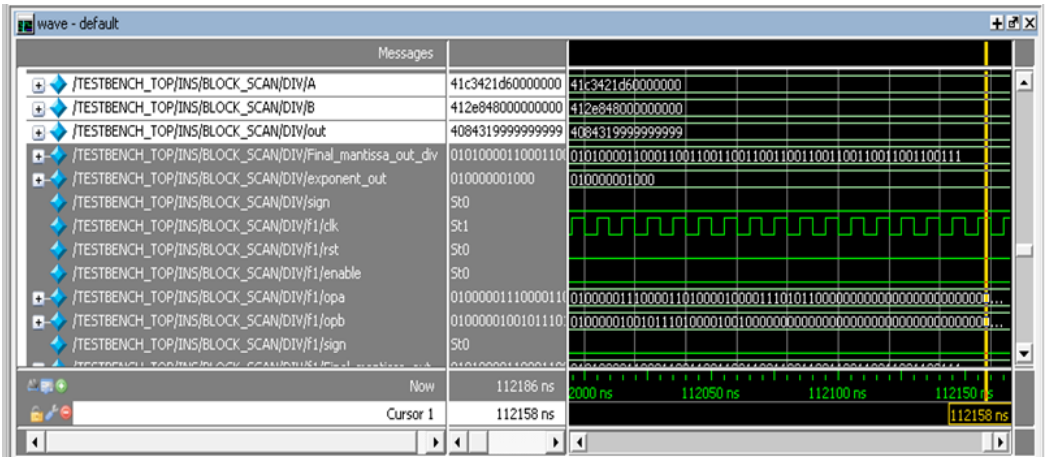


Figure 6. The floating point divider reconstructed the output

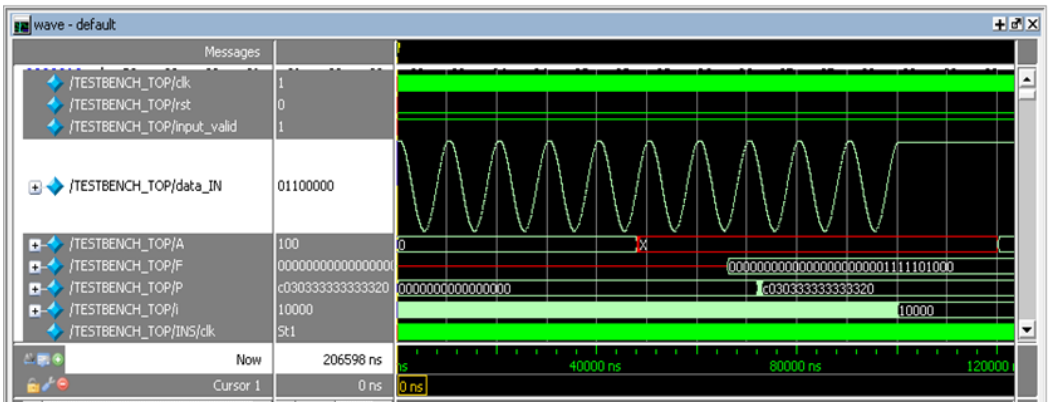


Figure 7. Phase estimation output

4. CONCLUSION

This article presents a comprehensive exploration of the FPGA-driven phase estimation with FPA, a pioneering approach in radar technology. The synergy of FPGA's reconfigurability and parallel processing capabilities, coupled with the precision of FPA, promises significant advancements in accuracy and efficiency. A comparative analysis with existing literature showcases the proposed approach's advantages in terms of area utilization, LUT, and DSP blocks. Despite a trade-off with delay, the lower power consumption suggests potential efficiency gains. The hardware implementation and experimental validation demonstrate the feasibility of real-time phase estimation at GHz rates. Results and discussions highlight the implementation details, hardware synthesis outcomes, and variable clock rate considerations. The presented comparative analysis provides a quantitative assessment of the proposed FPA-driven block-based phase estimator against existing literature.

ACKNOWLEDGEMENTS

The authors acknowledge the support and research facilities provided by the Department of Electronics and Communication Engineering, RVR&JC College of Engineering, Guntur, India.

REFERENCES

[1] S. Zereen, S. Lal, M. A. S. Khalid, and S. Chowdhury, "An FPGA-based controller for a 77 GHz MEMS tri-mode automotive radar," *Microprocessors and Microsystems*, vol. 58, pp. 34–40, Apr. 2018, doi: 10.1016/j.micpro.2018.02.005.

[2] H. Son, S. Lee, and K. Min, "FPGA implementation of UWB radar signal processing for automotive application," in *European Microwave Week 2010, EuMW2010: Connecting the World, Conference Proceedings-European Wireless Technology Conference, EuWiT 2010*, 2010, pp. 49–52.

[3] I. Firmansyah and Y. Yamaguchi, "FPGA-based implementation of a chirp signal generator using an OpenCL design," *Microprocessors and Microsystems*, vol. 77, Sep. 2020, doi: 10.1016/j.micpro.2020.103199.

[4] B. Steinmann, "Fixed point vs floating point arithmetic in FPGA," *imperix Knowledge base, TN148*, 2021. https://imperix.com/doc/implementation/fixed-point-vs-floating-point-in-fpga (accessed Dec. 10, 2023).

[5] M. Y. Chua *et al.*, "A miniature real-time re-configurable radar waveform synthesizer for UAV based radar," *Progress In Electromagnetics Research C*, vol. 31, pp. 169–183, 2012, doi: 10.2528/PIERC12060801.

[6] H. Huang *et al.*, "Improving Time-Efficiency of Variational Specific Differential Phase Estimation," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 7, pp. 5642-5664, July 2021, doi: 10.1109/TGRS.2020.3020914.

[7] Y. Choi, D. Jeong, M. Lee, W. Lee, and Y. Jung, "FPGA implementation of the range-doppler algorithm for real-time synthetic aperture radar imaging," *Electronics*, vol. 10, no. 17, Sep. 2021, doi: 10.3390/electronics10172133.

[8] Y. K. Chan, Y. C. Lee, and V. C. Koo, "Design and implementation of synthetic aperture radar (SAR) field-programmable gate array (FPGA)-based processor," *Applied Sciences*, vol. 12, no. 4, Feb. 2022, doi: 10.3390/app12041808.

[9] Y. Izumi, G. Nico and M. Sato, "Time-Series Clustering Methodology for Estimating Atmospheric Phase Screen in Ground-Based InSAR Data," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1-9, 2022, Art no. 5206309, doi: 10.1109/TGRS.2021.3072037.

[10] T. T. H. Kim, "Overview of memory design for next generation applications," in *Proceedings-2019 International SoC Design Conference, ISOCC 2019*, 2019, pp. 162–163, doi: 10.1109/ISOCC47750.2019.9027664.

[11] "NXP support package S32R," *MathWorks*. https://www.mathworks.com/matlabcentral/fileexchange/72232-nxp_support_package_s32r/ (accessed Dec. 07, 2023).

[12] S. Mukherjee, "On the radar," *Force India*. https://forceindia.net/guest-column/on-the-radar/ (accessed Dec. 07, 2023).

[13] Leonardo, "The future of radar: the evolution of a technology with a long history," *Leonardo*. https://www.leonardo.com/en/news-and-stories-detail/-/detail/the-future-of-radar-the-evolution-of-a-technology-with-a-long-history (accessed Dec. 07, 2023).

[14] C. Le *et al.*, "Onboard FPGA-based SAR processing for future spaceborne systems," in *IEEE National Radar Conference-Proceedings*, 2004, pp. 15–20, doi: 10.1109/nrc.2004.1316388.

[15] C. Lesnik, A. Kawalec, and P. Serafin, "A real time SAR processor implementation with FPGA," *WIT Transactions on Modelling and Simulation*, vol. 51, pp. 435–444, 2011, doi: 10.2495/CMEM110381.

[16] Y. Lou, D. Clark, P. Marks, R. J. Muellerschoen, and C. C. Wang, "Onboard radar processor development for rapid response to natural hazards," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 9, no. 6, pp. 2770–2776, 2016, doi: 10.1109/JSTARS.2016.2558505.

[17] C. Yang *et al.*, "A spaceborne synthetic aperture radar partial fixed-point imaging system using a field-programmable gate array-Application-specific integrated circuit hybrid heterogeneous parallel acceleration technique," *Sensors (Switzerland)*, vol. 17, no. 7, 2017, doi: 10.3390/s17071493.

[18] R. Liu, D. Zhu, D. Wang and W. Du, "FPGA Implementation of SAR Imaging Processing System," *2019 6th Asia-Pacific Conference on Synthetic Aperture Radar (APSAR)*, Xiamen, China, 2019, pp. 1-5, doi: 10.1109/APSAR46974.2019.9048543.

[19] K. H. Chen and Y. S. Chu, "A low-power multiplier with the spurious power suppression technique," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 15, no. 7, pp. 846–850, 2007, doi: 10.1109/TVLSI.2007.899242.

[20] M. Rupniewski, G. Mazurek, J. Gambrych, M. Nalecz, and R. Karolewski, "A real-time embedded heterogeneous GPU/FPGA parallel system for radar signal processing," in *Proceedings-13th IEEE International Conference on Ubiquitous Intelligence and Computing, 13th IEEE International Conference on Advanced and Trusted Computing, 16th IEEE International Conference on Scalable Computing and Communications, IEEE Internationa*, 2017, pp. 1189–1197, doi: 10.1109/UIC-ATC-ScalCom-CBDCom-IoP-SmartWorld.2016.0182.

[21] Y. Feng, S. Hu, X. Li, and J. Yu, "Implementation and optimisation of pulse compression algorithm on open CL-based FPGA," *The Journal of Engineering*, vol. 2019, no. 21, pp. 7752–7754, 2019, doi: 10.1049/joe.2019.0757.

[22] "Low power FPGAs and PLDs," *Microchip*. https://www.microchip.com/en-us/products/fpgas-and-plds/low-power (accessed Dec. 07, 2023).

[23] A. Eddla and V. Y. J. Pappu, "Low area and power-efficient fpga implementation of improved AM-CSA-IIR filter design for the DSP application," *International Journal of Electrical and Electronic Engineering and Telecommunications*, vol. 11, no. 4, pp. 294–303, 2022, doi: 10.18178/ijeetc.11.4.294-303.

[24] S. Ullah, S. Rehman, M. Shafique, and A. Kumar, "High-performance accurate and approximate multipliers for FPGA-based hardware accelerators," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 2, pp. 211–224, 2022, doi: 10.1109/TCAD.2021.3056337.

[25] P. Kwiatkowski, "Digital-to-time converter for test equipment implemented using FPGA DSP blocks," *Measurement: Journal of the International Measurement Confederation*, vol. 177, 2021, doi: 10.1016/j.measurement.2021.109267.

BIOGRAPHIES OF AUTHORS

Siva Prasad Ponduri received an engineer degree in electronics and Communication Engineering from, Chennai in 2001 and a Master in Digital Electronics and Communication Engineering from JNTU Hyderabad. He currently working as an assistant professor at the Department of Electronics and Communication Engineering, RVR&JC College of Vel Tech Engineering College Engineering, Guntur. He received the best teacher award in 2012 from the Rotary club of Guntur, Andhra Pradesh. He is currently doing Ph.D. at Visvesvaraya Technological University Belagavi, Karnataka-590018, India. He can be contacted at email: ponduri2002@gmail.com.

Anandi Venkataraman is an Associate Professor at the Department of Electronics and Communication Engineering, MSRIT, Bangalore, where she has been a faculty member since 1997. She graduated with a B.E. degree in Electronics and Communication Engineering from RMadhurai Kamaraj University in 1988 and an M.S. in Electronics and Communication Engineering from Birla Institute of Technology, Birla in 1997 and completed her Ph.D. Engineering from the Anna Universit, Chennai, in 2015. Her research interests are primarily in the area low power VLSI design and microelectronics. She can be contacted at email: anandi.v@msrit.edu.

P Satyanaryana Murty received his Engineer Degree in Electronics and Communication Engineering from K.B.N. College of engineering, Gulbarga University, Gulbarga in 1994, and the Masters in Digital Electronics and Communication Engineering from JNTU Hyderabad. He received his Ph.D. from Andhra University, Visakhapatnam in 2014. His research interests are primarily in the area signal and image processing. Presently he is working as Principal and Professor at the Department of Electronics and Communication Engineering, Gonna Institute of Information Technology and Science, Aganampudi, AnakapalleDist, A.P. He can be contacted at email: psn.murty71@gmail.com.