

## **Improved performance of sonobuoys based on open source RISC-V SoC microprocessor technology**

Vu Le Ha<sup>\*</sup>, Nguyen Manh Cuong, Phan Hong Minh

Institute of Electronics, Academy of Military Science and Technology, 17 Hoang Sam, Cau Giay, Hanoi, Vietnam.

<sup>\*</sup>Corresponding author: vuleha.vdt@gmail.com

Received 27 Oct. 2024; Revised 23 Dec. 2024; Accepted 25 Dec. 2024; Published 25 Feb. 2025.

DOI: <https://doi.org/10.54939/1859-1043.j.mst.101.2025.56-63>

### **ABSTRACT**

*This paper presents a System on Chip (SoC) design solution using Rocket core supporting open source RV32GC RISC-V instruction set, verifying the design on pre-silicon platforms which are Vivado FPGAs. The SoC system includes Rocket RISC-V core integrating peripheral blocks UART, SPI, RAM, ROM, GPIO, TILELINK BUS and a SonarDetect block (custom hardware developed by the user). This solution is applied to replace the central processor of the PTA-18 sonobuoy [1], develop and upgrade the target identification feature compared to the old design. SonarDetect is a new user design block (Custom Hardware) that detects the signals of local sonar targets, compares with the given database to identify the target. The design is tested and implemented on the Arty-A7 100T FPGA development kit and aims at the next goals of smartening and chip-izing the buoy control circuit.*

**Keywords:** Risc-V Rocket; SoC; PTA-18; Arty-A7100T; Chisel; Acoustic Identification.

### **1. INTRODUCTION**

Reduced Instruction Set Computer - version five architecture (RISC-V) [2, 3] is an open source Instruction Set Architecture (ISA) designed to be modular, extensible, and highly customizable. The architecture is not subject to copyright restrictions and can be further developed to integrate specialized functional modules, making it an attractive choice for custom-designed systems. Modularity and flexibility are strong features of this microprocessor system. In addition, RISC-V adheres to a synchronous and sequentially consistent memory model, which simplifies programming and porting code across different implementations. Furthermore, the architecture provides optimizations for low power consumption by optional extensions and configurations [4]. The architecture also provides a flexible platform for creating custom processor cores, especially SoC platforms using RISC-V CPU on FPGAs that are suitable for many types of projects such as: embedded systems on chips for electronic devices, real-time applications and AI accelerators, which are now widely used in many applications and fields of the electronics industry. On that basis, this paper presents a RISC\_V SoC hardware design solution on FPGA platform towards replacing the old control hardware of PTA18 buoy. In part 2 of the paper, in addition to the new system hardware solution, the buoy's functionality is also upgraded through a new IP block called Sonar Detect with a new algorithm to perform target recognition. The results of the hardware solution implementation and target recognition function are presented in section 3. The new hardware solution in the direction of SoC using RISC-V on the FPGA platform has overcome the limitations of the old design and opened up the possibility of smart and chip-ization for sonobuoys and realized the goal of effectively applying new technology and research into practice.

### **2. BUILDING STRUCTURES AND SOLUTIONS**

#### **2.1. Building a new FPGA-based RISC-V SoC architecture for PTA-18 sonobuoy**

The PTA18 sonobuoy is responsible for detecting underwater targets through acoustic signals. The hardware design of the PTA18 central processor uses ARM32-STM controllers and separate functional ICs, implementing the algorithm flowchart according to figure 1. The buoy's function is

to detect acoustic signals in the working frequency range and collect data and transmit them to the central station on shore with warning signals. The buoy is not currently integrated with digital signal processing cores such as IP cores, DSP cores, AI engines,... used to detect and identify underwater targets on the spot, leading to limitations in slow recognition processing speed, long data transmission time to the center, leading to high energy consumption and low transmission reliability.

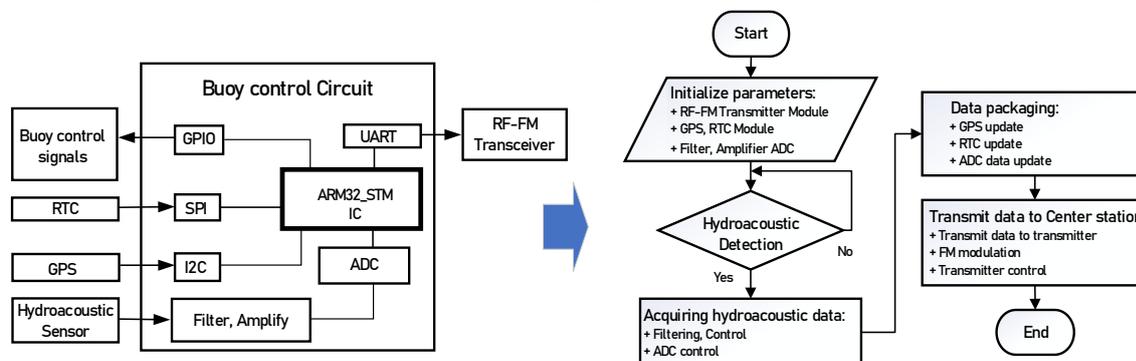


Figure 1. Block diagram and algorithm of PTA-18 sonobuoy control with control circuit from Ics.

The new SoC architecture (figure 2) aims to replace the PTA18 sonobuoy control circuit, with the goal of upgrading features, performance, reliability, minimizing hardware complexity and manufacturing costs of the buoy. The functional peripheral ICs are replaced by open source IP cores, the ARM32\_STM processor IC is replaced by the Rocket CPU RISC-V core. The fundamental change of the SoC system compared to the old control circuit is that the Sonar Detect hydroacoustic signal processing block has been integrated as a user IP to take advantage of the resources and powerful computing capabilities of the FPGA. In the old design, hydroacoustic data was transmitted to the central station for processing, leading to longer processing time, and long data transmissions consumed more energy. With the new design, on-site computing has significantly improved the above disadvantages.

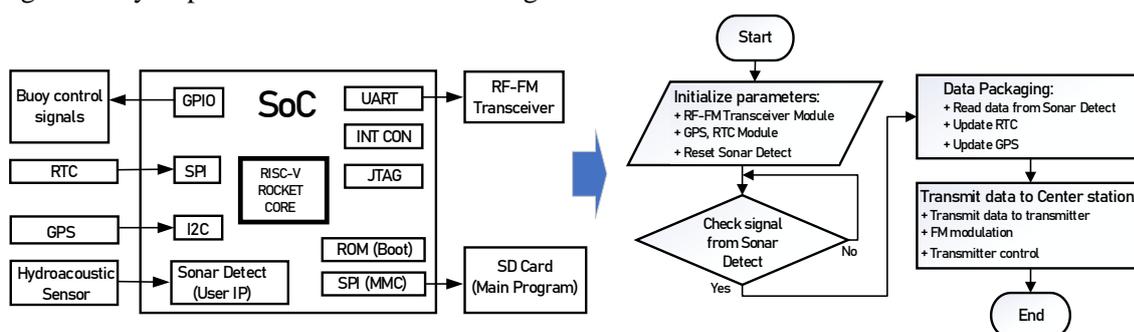


Figure 2. 18 sonobuoy control algorithm.

The SoC system architecture (figure 3) is designed based on the Rocket Chip SoC model [5][6], which resembles a simple computer system with a Rocket RISC-V CPU core as the central processor. In the design of the SoC for the sonobuoy, a 32-bit Rocket core supporting the I,M,A,F,D,C instruction set extensions (referred to as RV32GC) is used in combination with peripheral IP blocks to communicate with the sonobuoy hardware functional modules such as GPS, environmental sensors, and radio data transceivers.

The peripheral blocks (UART, I2C, SPI,...) in the design are open source IPs. The Sonar Detect block is written in Verilog language with the function of receiving sonar signals from the sensor, analyzing and comparing with saved samples and giving evaluation results. The SoC system will

be connected to the buoy's functional modules as shown in figure 2. The functional blocks operate and provide information about sonar targets, GPS positioning, and real-time information to the Rocket central processor to synthesize information and push data to the RF-FM transmitter module to transmit messages to the center.

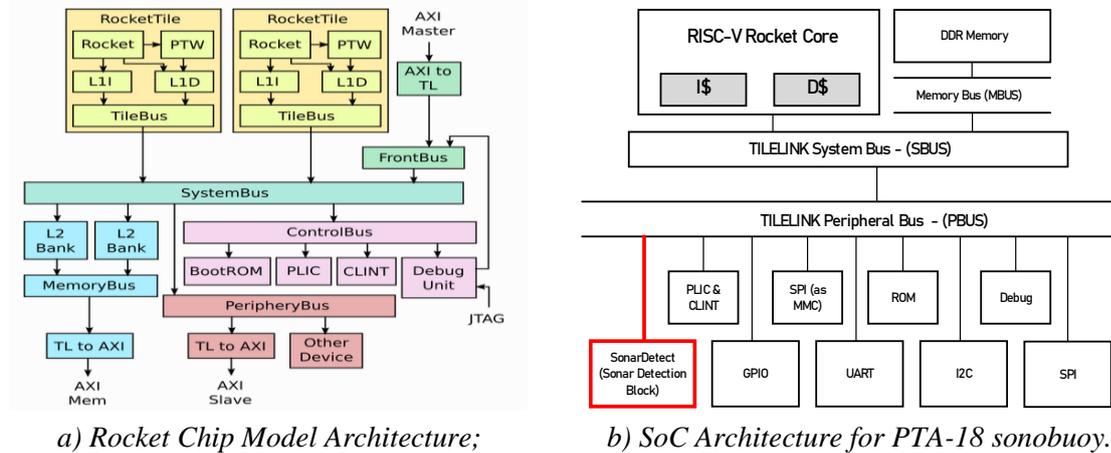


Figure 3. RISC-V SoC architecture underlying the PTA-18 sonobuoy.

2.2. Design of user IP custom hardware block - Sonar Detect

2.2.1. Block diagram

The IP SonarDetect block is a custom hardware block for the task of detecting strange targets outside the stored catalog. The task of the block is to compare the received sample with the samples in the library to give an alert, if the received signal is from unknown sources (unsaved samples).

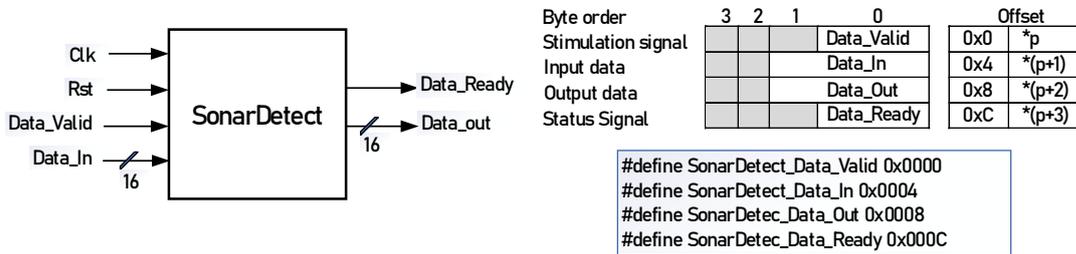


Figure 4. Sonar Detect block diagram and register mapping to hardware memory map.

Whenever data comes from the sonar sensor, the Data\_Valid signal is activated for the block to work, during the calculation block, the Data\_Ready signal will be deactivated, when the calculation block is finished, the hardware will reactivate Data\_Ready and report the results are ready at Data\_Out. The main program reads the target classification results and gives warnings, figure 4 describes the block diagram of Sonar Detect and maps the related registers to the memory map of the SoC hardware.

2.2.2. Target database and signal characteristics

Hydroacoustic targets are sound signals emitted from natural sources and artificial sound sources created by human activities. These sounds have their own frequency and timbre characteristics, from which they can be classified [7]. The paper uses a number of underwater sound sources to test and evaluate the design solution and the effectiveness of the algorithm.

2.2.2.1. Submarine signals

When operating, submarines emit many types of characteristic underwater signals such as: Ping sound for communication, propeller rotation sound, engine sound and auxiliary system sound.

Submarine sounds are taken from the Soundsnap sound library [8], using Matlab tool to analyze and evaluate characteristic frequency and signal envelope spectrum figure 5 and table 1 as the database for the recognition algorithm of the Sonar Detect block.

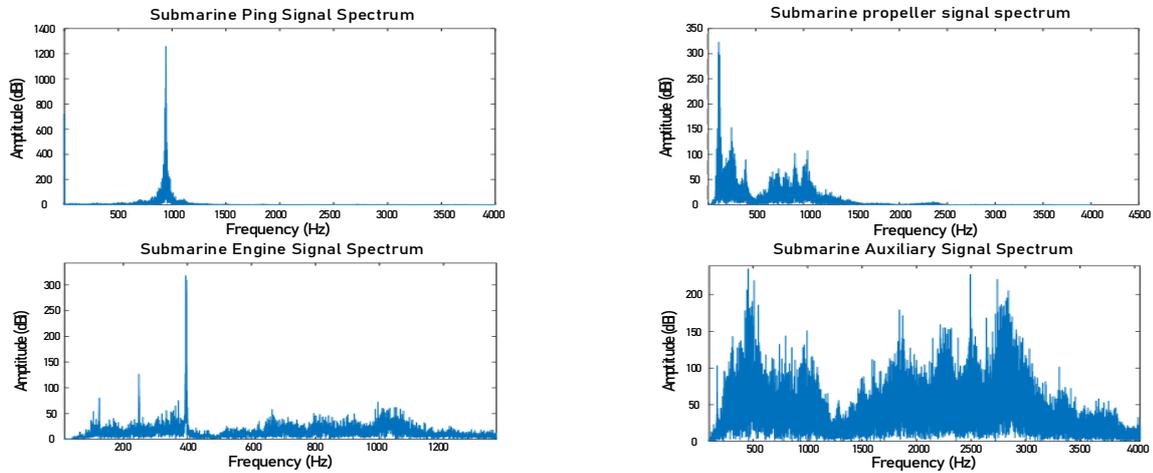


Figure 5. Frequency spectrum of submarine signals.

Table 1. Submarine acoustic signal characteristic parameter table.

Submarine signal type	Characteristic frequencies (Hz)
Ping sound	1000
Propeller	100, 250, 1000
Main engine	250, 400
Auxiliary engine	250, 490, 1100, 1800, 2500,

2.2.2.2. Signals of some types of civil ships

The paper also uses some underwater sound sources directly collected from civil vessels such as high-speed canoes, passenger ships and transport ships to create a real database for the Sonar Detect block. Each signal received will be compared with the samples in the database to distinguish whether it is strange or not. The signal characteristic analysis is shown in figure 6 and table 2.

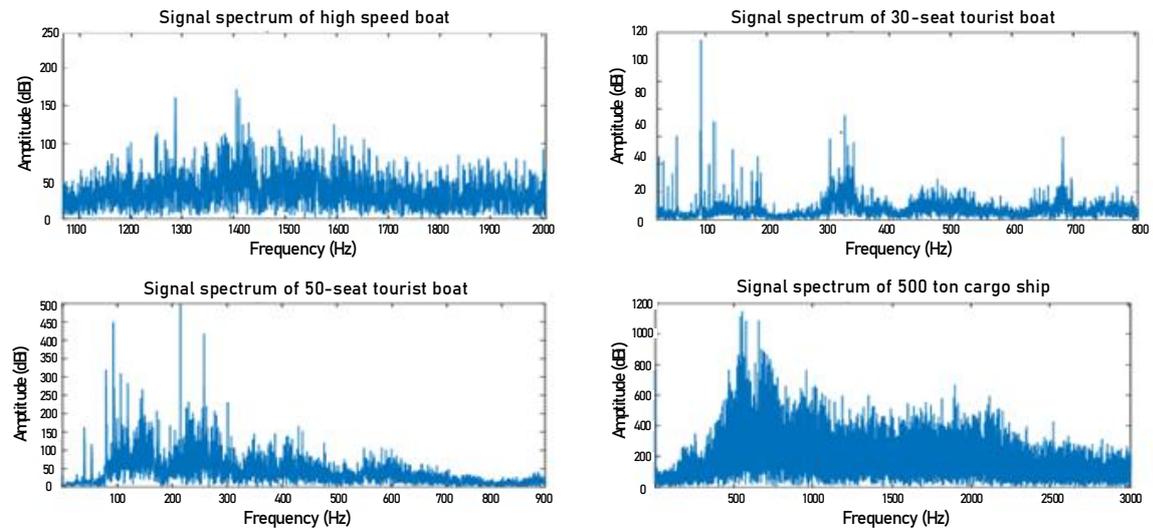


Figure 6. Frequency spectrum of civil ship signals.

Table 2. Table of signal characteristics parameters of civil ships.

Civilian ship type	Characteristic frequencies (Hz)
Speedboat	1289, 1406
30-seat tourist boat	53, 93, 114
50-seat tourist boat	93, 215, 252, 259
500 ton cargo ship	541, 552, 554 and 568

2.2.3. Signal processing algorithm

The audio signal is sampled at 8 KHz frequency with 8 bit resolution and will be represented by data samples  $S(i)$ ,  $i = 1, \dots, n$  are bit clusters with specific values carrying information about the amplitude and phase of the signal. To calculate the signal energy in a bit cluster with a length of 8 samples corresponding to 4 ms of the incoming signal according to the expression:

$$\Delta W_i = \sum_{i=1}^{N=8} |S(n)|^2 \tag{1}$$

Use the discrete Fourier transform (DFT) and inverse transform (IDFT) algorithms for the audio samples according to the expressions (2)(3) to determine the characteristic frequencies of the audio emitted from the target [9, 10].

$$Y(k) = \sum_{j=1}^n X(j)W_n^{(j-1)(k-1)} \tag{2}$$

$$X(j) = \frac{1}{n} \sum_{k=1}^n Y(k)W_n^{-(j-1)(k-1)}, \text{ where } W_n = e^{(-2\pi j)/n} \tag{3}$$

With 4 ms signal samples, consider the 4 s time period of the received signal to obtain 1000 energy value samples corresponding to the signal frequencies. Compare these samples with the target database to classify with the corresponding ratio.

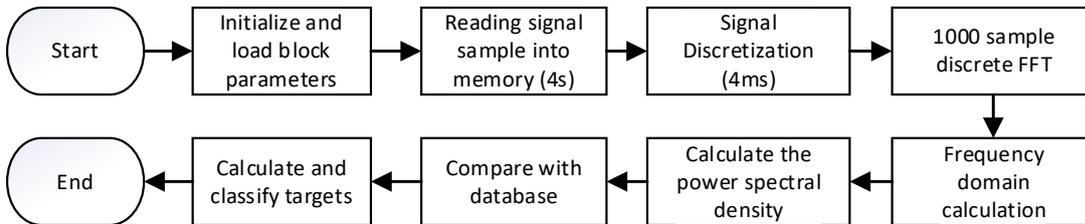


Figure 7. Algorithm flowchart.

3. RESULTS AND DISCUSSION

3.1. Hardware implementation results

The SoC system hardware implementation is done through the Chisel hardware architecture language [11]. Chisel is an open source hardware description language (HDL), developed from the embedded language Scala, used to describe electronic circuits and digital electronic devices at the register translation level.

In the SoC design in this paper, the peripheral blocks of the system are represented in the form of Chisel libraries, the use is supported in the form of optimized variables or functions, just need to declare the correct format to suit the usage target. The connection and configuration of the system (the parameters of the Rocket core, declaration of peripheral IPs or memory address space, connection to the TiteLink system bus) according to the design uses scalar configuration files for description. Sonar Detect is a user hardware block (User Hardware) integrated into the SoC system

using the TileLink bus protocol via BlackBox [12].

After the configuration with Chisel/Scalar is completed, the system is compiled through various steps and tools. The scalar files are compiled through the Build.Sbt tool into (.fir) files, the FIRRTL tool compiles the .fir files into Verilog (.v) files, and finally the Vivado tool compiles the Verilog files into bitstreams (.bit) so that the SoC execution can be downloaded to the FPGA.

Total LUTs	FlipFlops	RAMB36	RAMB18	DSP48
23171	16347	7	8	2

Figure 8. FPGA resource usage results of the SoC system after compilation.

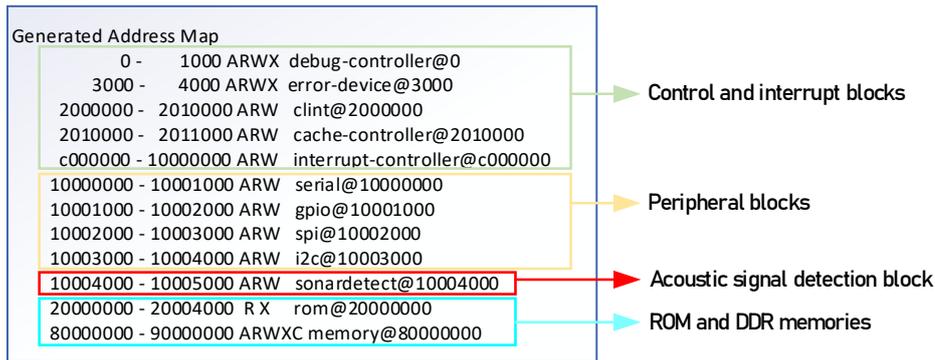


Figure 9. SoC hardware compilation results in the Device Tree .dts output file.

The boot program and sonobuoy control on the SoC system hardware platform will be compiled into assembly code by the RISC-V toolchain [13, 14] for execution by the RISC-V CPU core.

### 3.2. Test to evaluate target recognition results

In the laboratory solution evaluation test, all 8 hydroacoustic data samples in section 2 are used as shown in table 3. Five samples are saved to the SD card to make a database of known samples (Sample 001 to 005) and 3 samples 6, 7 and 8 are used as strange samples. The testing process will use the above 8 hydroacoustic sound samples to play on the hydroacoustic sensor for the SonarDetect block and the SoC system to work. When playing the samples saved in the system, the buoy system will recognize and evaluate the signal similarity ratio, and for samples that are not saved, the system will report that it cannot be recognized. Each received signal has a fixed length of 4 s, the spectrum is analyzed and linearized to 1000 samples, compared with the frequency of the signals in the database and calculates the similarity ratio. If the sample similarity ratio is below the preset threshold (e.g. 50%), the system will report Unidentified and trigger an alarm. If the similarity ratio is greater than 50%, the system will report the sample is recognized and the signal similarity ratio (figure 10).

Table 3. Hydroacoustic samples used for laboratory evaluation.

Sample	Hydroacoustic data	Database Name
Sample 1	High Speed Boat	Sample 001
Sample 2	Submarine Ping Signal	Sample 002
Sample 3	Submarine propeller signal	Sample 003
Sample 4	Submarine main engine	Sample 004
Sample 5	30-seat tourist boat	Sample 005
Sample 6	50-seat tourist boat	No Save (Strange Pattern)
Sample 7	Submarine auxiliary engine	No Save (Strange Pattern)
Sample 8	500 ton cargo ship	No Save (Strange Pattern)

```

daobaanh@ubuntu: ~
File Edit View Search Terminal Help
CMD16
/ 80078200 <- 00000782kB / 00000800kB
          BOOTING RATONA:

RATONA PTA18 Sonar Detect Buoy:      2024-10-22-__TIME__:07:29:03.
Got TL_CLK: 50000000
Got RISC_V ROCKET NUM_CORES: 1
Got TIMEBASE: 1000000
----- Sonar Detect Log-----
--Signal 1 -- Fitting: Sample-003/80% -- Time: 07:35:11 -- Date: 22/10
--Signal 2 -- Fitting: Sample-005/91% -- Time: 07:38:22 -- Date: 22/10
--Signal 3 -- Fitting: Unidentified   -- Time: 07:41:09 -- Date: 22/10
--Signal 4 -- Fitting: Sample-001/61% -- Time: 07:47:44 -- Date: 22/10
--Signal 5 -- Fitting: Sample-001/69% -- Time: 07:55:26 -- Date: 22/10
--Signal 6 -- Fitting: Sample-001/94% -- Time: 07:59:06 -- Date: 22/10
--Signal 7 -- Fitting: Sample-004/87% -- Time: 08:05:36 -- Date: 22/10
--Signal 8 -- Fitting: Sample-002/96% -- Time: 08:41:09 -- Date: 22/10
--Signal 9 -- Fitting: Sample-002/85% -- Time: 08:46:19 -- Date: 22/10
--Signal 10 -- Fitting: Unidentified  -- Time: 08:50:47 -- Date: 22/10
CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.7.1 | VT102 | Offline

```

**Figure 10.** Laboratory system testing results.

After combining the Arty100T on-board SoC system with the sonar sensor and collecting sonar samples in the laboratory, the results were achieved as shown in figure 10. The samples were transmitted and recognized with a high similarity rate. Strange samples were received at the 3rd and 10th transmissions and the system detected and reported that they were not recognized. During the testing process, we also changed the samples and databases many times, changed the transmission power levels (Example: Samples-001 and 002 in figure 10), some adjustments to the sampling rate and processing algorithm were also adjusted to achieve the highest recognition similarity rate of the SonarDetect block.

#### 4. CONCLUSIONS

The paper presents the implementation of a system-on-chip design that applies the RISC-V Rocket open source processor platform using advanced design tools today. The new platform uses open source IPs, implemented on the FPGA platform, allowing easy optimization of applications, reusing programming code, leading to cost-effective investment and development time for application products such as the PTA18 sonobuoy compared to the previous design method from discrete functional ICs. In addition, the paper also presents a new central processor design solution that integrates a signal detection block for the PTA18 sonobuoy by building a Sonar Detect IP block integrated into the SoC, built to process sonar signals on-site instead of transmitting data to the center for processing. The new solution takes advantage of the speed and processing storage capacity of the FPGA platform to improve the features and performance of the buoy while simplifying the hardware design because the functional blocks are implemented by soft core IPs. Based on these results, the next research direction of the paper is to develop AI-ized hydroacoustic signal recognition blocks to identify multiple targets and classify more accurately, automatically save samples and classify, and further, to be able to chip-ize the application when the product is put into production after research and development.

#### REFERENCES

- [1]. Le Ky Bien, Phan Hong Minh, Tran Hieu Thao, Phan Trong Hanh, “ *Signal processing solution for sonobuoy system to detect and warn targets based on passive sonar principle* ”, Journal of Military Science and Technology Research, Special Issue FEE, pp. 95-105, (2018).
- [2]. Andrew Waterman, Krste Asanovi, and Five Inc. “ *The RISC-V Instruction Set Manual Volume I: User-Level ISA Document Version 2.* ” CS Division, EECS Department, University of California, Berkeley (2017).

- [3]. David A. Patterson and John L. Hennessy, "Computer Organization and Design RISC-V Edition: The Hardware Software Interface." Page 22 Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, (2017).
- [4]. Yuhao Xi, Zhendong Zhang, Yuze Wang, Wen Wang, Xiaoxia Han, Weidong Wang, Peng Liu, "A Heterogeneous RISC-V SoC for Confidential Computing and Password Recovery", The 7th International Conference on Integrated Circuits and Microsystems, pp. 500-504, (2022). DOI: 10.1109/ICICM56102.2022.10011247.
- [5]. <https://chipyard.readthedocs.io/en/stable/Generators/Rocket-Chip.html>
- [6]. <https://github.com/chipsalliance/rocket-chip>
- [7]. Phan Hong Minh, Phan Van Viet, Nguyen Thi Thu Hong, Pham Thanh Thuong, "A solution to create customized beams for three-dimensional circular arrays of 1H antennas in hydroacoustic sonar systems MFK-400EM", Journal of Military Science and Technology Research, Special Issue of the Institute of Electronics, pp. 227-239, (2020).
- [8]. <https://www.soundsnap.com/search/audio?query=submarine&page=2>
- [9]. Qihu Li, "Digital Sonar Design in Underwater Acoustics, Principles and Applications", Zhejiang University Press, Hangzhou and Springer-Verlag Heidelberg, pp.43-45, (2012).
- [10]. Frigo, M., and S. G. Johnson. "FFTW: An Adaptive Software Architecture for the FFT." Proceedings of the International Conference on Acoustics, Speech, and Signal Processing. Vol. 3, pp. 1381-1384, (1998).
- [11]. Bachrach, J.; Vo, H.; Richards, B.; Lee, Y.; Waterman, A.; Avižienis, R.; Wawrzynek, J.; Asanović, K. "Chisel: constructing hardware in a Scala embedded language". Proceedings of the 49th Annual Design Automation Conference (DAC 2012). San Francisco, California, US: Association for Computing Machinery (ACM), pp.1216-1225, (2012). doi:10.1145/2228360.2228584. ISBN 978-1-4503-1199-1.ABSTRACT
- [12]. <https://www.chisel-lang.org/docs/explanations/blackboxes>
- [13]. <https://riscv.org/wp-content/uploads/2015/01/riscv-software-toolchain-workshop-jan2015.pdf>.
- [14]. <https://www.linkedin.com/pulse/overview-risc-v-toolchain-sibraintech>.

## TÓM TẮT

### Cải thiện hiệu năng của phao thủy âm vô tuyến dựa trên công nghệ vi xử lý mã nguồn mở RISC-V SoC

Bài báo này trình bày một giải pháp thiết kế vi mạch SoC sử dụng lõi Rocket hỗ trợ bộ lệnh RV32GC RISC-V mã nguồn mở, kiểm chứng thiết kế trên các nền tảng tiên silicon là các FPGA Vivado. Hệ SoC bao gồm lõi RISC-V Rocket tích hợp các khối ngoại vi UART, SPI, RAM, ROM, GPIO, TILELINK BUS và một khối SonarDetect (phần cứng tùy chỉnh do người dùng phát triển). Giải pháp này ứng dụng để thay thế bộ xử lý trung tâm của phao phát hiện thủy âm vô tuyến PTA-18 [1] và phát triển nâng cấp tính năng nhận dạng mục tiêu so với thiết kế cũ. SonarDetect là một khối thiết kế người dùng mới (Custom Hardware) làm nhiệm vụ phát hiện tín hiệu các mục tiêu thủy âm tại chỗ, so sánh với cơ sở dữ liệu cho trước để nhận dạng mục tiêu. Thiết kế được kiểm thử, thực thi trên bộ kit phát triển FPGA Arty-A7 100T và hướng đến các mục tiêu tiếp theo là thông minh hóa và chip hóa cho phần mạch điều khiển phao.

**Từ khóa:** Risc-V Rocket; SoC; PTA-18; Arty-A7100T; Chisel; Nhận dạng thủy âm.