

# AI-Accelerated Heterogeneous Multi-Core SoC for Real-Time ECG Analysis

Subbulakshmi A

Assistant Professor

Department of Electronics Engineering

(VLSI Design and Technology)

Chennai Institute of Technology

Chennai, Tamil Nadu, India

subbulakshmiarunachalam12@gmail.com

Balaji S

Electronics and Communication  
Engineering

Chennai Institute of Technology

Chennai, Tamil Nadu, India

balajis07032005@gmail.com

Manikandan L

Electronics and Communication  
Engineering

Chennai Institute of Technology

Chennai, Tamil Nadu, India

manikandan23305@gmail.com

**Abstract**—In this paper, the design of a heterogeneous multi core system on chip will be described and will be used in real time biomedical data analysis. Most current healthcare monitoring systems rely on cloud computing or general-purpose processors and hence may cause delays, increased power usage, and data privacy issues. A RISC V based architecture that introduces the processing capability onto the chip is proposed in this work. It has a general-purpose core of control facilities, and a second processing core, optimized to do digital signal processing. New custom hardware instructions are added to accelerate these common biomedical signal processing filtering and frequency analysis operations. There is also a hardware convolutional neural network accelerator to assist in automatic diagnosing the device. Since all computation will be done on-site, the system will be quicker and will not expose sensitive medical information to external networks. The design suggested demonstrates that a tailored hardware that is developed on an open processor platform could offer an effective and secure means of continuous biomedical monitoring appliances.

**Keywords**—*Biomedical signal processing, RISC V, multi core processor, hardware accelerator, real time monitoring, convolutional neural network*

## I. INTRODUCTION

Biomedical monitoring systems are quite important in the contemporary medical care particularly due to the rising demand of constant and real time monitoring of the patient. Signals like the electrocardiogram are useful in clinical diagnosis and the signal is usually compromised due to noise thus needs to be processed promptly and correctly before a diagnosis can be arrived at. Most of the existing systems use cloud services or commodity processors to do this calculation. Although they are convenient, they can create delays in the processing, consume more power, and bring up privacy of sensitive medical data. Such constraints complicate the realization of reliable real time performance of portable or embedded healthcare devices. In order to address these issues, an increased focus is being given to the design of specific hardware platforms optimized to the analysis of biomedical signals. Those systems could have direct computation inside the device, which minimizes latency and makes sure that patient data does not leave the device. This paper will design a custom system on chip architecture using the RISC V architecture. The design employs heterogeneous multi core design, with one of the cores being used to carry out general control tasks and the other core being specifically designed to do the tasks of a digital signal processing. Additional hardware assistance is added to hasten activities like filtering, frequency

examination and automated pattern identification. This architecture has the purpose of offering a secure, efficient and scalable platform to continual biomedical monitoring. The integration of open processor technology with customization of application specific hardware depicts a viable course of action to more rapid and dependable medical signal examination in real time settings as proposed system.

## II. SYSTEM ARCHITECTURE AND COMPONENTS

The proposed System on Chip incorporates some of the major hardware blocks which collaborate to facilitate the real time biomedical signal analysis. The Finite Impulse Response (FIR) filter module is one of the fundamental components that are employed to eliminate noise and other unwanted effects of raw biomedical signals in a form of ECG waveforms [21] [2]. These signals are easily influenced by movement, electrical noise, and other factors in the environment; thus, a good filtering is required so that the important diagnostic information is not lost. Application of FIR filter in hardware enables the system to purify the data stream that is coming in, with minimal delay and reduced power consumption compared to a typical software-based method. The system has a special Fast Fourier Transform (FFT) processing unit in addition to filtering. This module transforms the signal in time domain to the frequency domain and this allows it to examine the spectral make-up of the waveform and find concealed frequency content or repetitive noise patterns [18] [4]. The frequency analysis is influential especially in biomedical monitoring whereby it assists in identifying irregularities and prepares the data to be subjected to further interpretation by machine learning [21] [10]. Performing FFT operations in hardware directly, the SoC can efficiently perform them and provide quick and efficient calculations befitting continuous real time work [18] [13]. Each module is assigned a specific functional role within the processing pipeline, as summarized in Table I. The physical organization of these components within the SoC is illustrated in Fig. 1.

The processor architecture is extended to add custom extensions to the standard instruction set in order to further improve performance. They are written to target computationally intensive digital signal processing operations (like multiply-accumulate operations and data rearrangement) [1]. The tailored hardware instructions enable them to be executed in one cycle as opposed to performing them with numerous software instructions. This saves execution time, decreases energy usage and illustrates the advantage of processor architecture customization to

application like biomedical analytics. Lastly, the system incorporates a hardware-based Convolutional Neural Network (CNN) accelerator of automatic diagnosis and pattern recognition [17]. Once the signal has been filtered and analyzed, the CNN then runs the data and identifies potential irregularities, which may be attributed to medical conditions. Neural network inference on the SoC does not require any information to be sent to external servers, increasing the speed and data privacy [4]. These elements combine to develop a safe, effective and extremely reactive platform of real time biomedical monitoring [20] [17] [2].

TABLE I: HETEROGENEOUS MULTI-CORE SOC COMPONENTS

Module	Description	Implementation
RISC-V Processor	Controls data flow	Custom RTL
FIR Filter	Noise removal	HW Accelerator
FFT Engine	Frequency analysis	HW Accelerator
CNN Engine	ECG classification	Custom RTL
Memory	Data buffering	On-chip SRAM



Fig. 1. System Architecture

### III. CUSTOM RISC-V INSTRUCTION SET ARCHITECTURE

The System on Chip is designed based on RISC V instruction set architecture, which has the capability of adding new instructions to fit into a particular application. Here, the basic RV32I pipeline is expanded to include extra hardware logic to permit custom instructions that are needed by biomedical signal processing [1]. The processor is based on a pipelined execution pattern, in which each instruction is processed in a fetch, decode, execute, memory, and write-back sequence. This design enhances throughput since it enables a series of instructions to be executed at the same time [13]. The custom instructions are part of the decode and execute phases in such a way that specialized digital signal processing instructions can be run in hardware with no impact on the normal instruction format. Consequently, the processor is also compatible with traditional RISC V software development tools and yet manages to provide application-specific acceleration [25] [1]. The overall block-level organization of the processor, including the custom instruction execution data path's, is shown in Fig. 2.

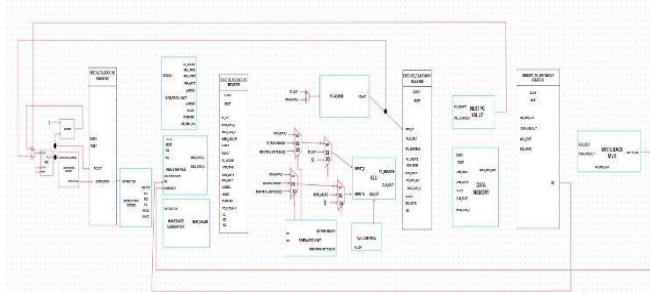


Fig. 2. Custom RISC V Architecture

#### A. VMAC Instruction for FIR Filter

The Vector Multiply-Accumulate (VMAC) instruction is one of the major custom instructions that have been adopted in the processor. The code achieves an accelerated Finite Impulse Response filtering that can be applied to get rid of noise in biomedical signals [1]. During the traditional processor, an FIR filtering means doing a series of multiply and addition operations many times in a row with multiple instructions, a cause of higher latency and higher power consumption. VMAC instruction can implement such operations in just one cycle of execution, and the result is accumulated in hardware by multiplying the input samples and the filter coefficient values. This goes a long way in cutting down the overall clock cycles needed to filter the signal and enables the system to handle biomedical signals at real times more efficiently [12].

#### B. Bit Reversal Instruction for FFT

The other significantly important extension is Bit-Reversal instruction that facilitates computation of Fast Fourier Transform. FFT algorithms make use of the use of bit-reversal addressing to rearrange input data prior to frequency-domain processing [2]. When done in pure software, this data rearrangement involves several memory operations and arithmetic operations, and this necessarily adds to the overhead. With the addition of dedicated Bit-Reversal instruction, the processor then can produce reordered indices using hardware logic and task is accomplished in a significantly shorter manner [24]. This makes the FFT pipeline parallel and the frequency based biomedical analysis more responsive, which makes the architecture applicable in continuous monitoring and high diagnostic rate applications [3].

The combination of these custom instructions, with matching hardware extensions on top of an open RISC V architecture, can achieve quite significant performance improvements without sacrificing architectural simplicity or compatibility [14].

### IV. FIR FILTER ARCHITECTURE AND WORKFLOW

The Finite Impulse Response (FIR) filter is another significant component in the proposed System on Chip and this filter helps in removing noise on the biomedical signal after which another processing can be undertaken [10]. Signals including electrocardiogram are prone to motion artifacts, electrical interference, and background disturbances, which disfigure features of the clinically significant waveforms. To avoid that, FIR filter is applied as a special hardware functionality to enable noise suppression to occur in real-time and without interruption [6]. This is for the reason that at the subsequent processing steps, the signal is left clean and acceptable in real time medical monitoring. Fig. 3 demonstrates the way to organize the FIR filtering block and to combine it with the processor using the custom VMAC instruction.

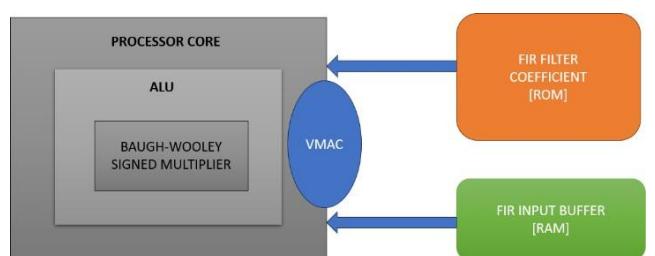


Fig. 3. FIR Filter Architecture

### A. Functional Role and Data Flow

The key objective of FIR filter is to tamper away the undesired frequency contents in the signal and to retain the valuable diagnostic information in the biomedical signal. Each new sensor sample is also added during operation to a buffer which contains a window of recent data points. Filter coefficients multiply these stored values and the co-products are added up to produce the filtered output [23] [12]. This is repeated with each incoming sample as a result of which the filter is operated in a completely streaming fashion. Since the calculation is performed on hardware, filtering is also done fast with less latency than when done in a software-based implementation [4].

### B. Instruction Level Acceleration using VMAC

To further make the computation more efficient, the multiply-accumulate address of the FIR filter are carried out using the custom Vector Multiply Accumulate (VMAC) instruction that is incorporated into RISC V processor [6]. The VMAC instruction does not require various sequential instructions to do the multiplication and addition operations, but does them in one hardware cycle. This allows cutting the time taken to execute and power usage down to a minimum, letting the FIR filter keep running without interruption to the real time stream of biomedical data. This leads to clean and stable signal generated by the system which can directly be forwarded to the FFT and CNN processing steps which can contribute to higher diagnostic accuracy and system overall performance [23] [17]. The register and memory activity that is produced upon VMAC-assisted execution is indicated in Fig. 4.

```

235 [REG] WRITE[3] : 0
245 [REG] WRITE[4] : 1025
255 [REG] WRITE[2] : 0
265 [REG] WRITE[3] : 0
275 [REG] WRITE[4] : 4102
275 [DATAMEM] WRITE[12288] : 4102
295 [REG] WRITE[4] : 0
305 [REG] WRITE[2] : 4102
315 [REG] WRITE[3] : 4102
325 [REG] WRITE[4] : 5126
335 [REG] WRITE[2] : 0
345 [REG] WRITE[3] : 0
355 [REG] WRITE[4] : 1025
365 [REG] WRITE[2] : 0
375 [REG] WRITE[3] : 0
385 [DATAMEM] WRITE[12288] : 6153
385 [REG] WRITE[4] : 6153
405 [REG] WRITE[4] : 0
415 [REG] WRITE[2] : 6153
425 [REG] WRITE[3] : 6153
435 [REG] WRITE[4] : 7177
445 [REG] WRITE[2] : 0
455 [REG] WRITE[3] : 0
465 [REG] WRITE[4] : 1025
475 [REG] WRITE[2] : 0
485 [REG] WRITE[3] : 0
495 [REG] WRITE[4] : 8204
495 [DATAMEM] WRITE[12288] : 8204
515 [REG] WRITE[4] : 0
525 [REG] WRITE[2] : 8204

```

Fig. 4. FIR Filter Data Log

## V. FFT ARCHITECTURE AND WORKFLOW

Another significant processing unit of the proposed System on Chip is the Fast Fourier Transform (FFT) module. The FIR filter removes noise in the time domain of the raw biomedical signal, but the FFT transforms the signal to the frequency domain to be able to detect the spectral aspects concealed within the signal. It is of particular use in biomedical measurement, where some abnormalities and repetitive distortions can be more readily identified based on frequency analysis than directly based on the raw waveform. The application of the FFT in hardware guarantees that

this computation can be done on-going even as the real time system runs.

### A. Functional Role and Data Flow

Operation involves the collection of blocks of filtered signal samples and sending it into the FFT processing unit. The signal is then broken down into the discrete frequency content to enable the system to view periodic patterns, noise characteristics and spectral patterns that are also of interest in the diagnosis. The CNN accelerator can also be fed on this frequency-domain information and classified. As this is calculated by a special hardware block the overall latency is significantly decreased when compared to running FFT fully in software. This enables the system to enable continuous and responsive biomedical monitoring.

### B. Instruction Level Acceleration using VMAC

To further improve the performance of FFT, the processor includes FFT-specific Bit Reversal instruction that helps in reordering of data needed by the FFT algorithm. This step in a traditional implementation has several memory reads and mathematical operations, thereby raising the processing time [24] [14]. When the instruction is dedicated, it is index reordering which is directly carried out in hardware and the FFT engine can continue faster with its calculation's steps. This decreases the total workload to execute the frequency analysis and makes sure that frequency analysis can match the rate of the incoming stream of biomedical signals. Consequently, the system can provide frequency-domain at a fast rate, providing effective and prompt medical evaluation.

## VI. CNN ACCELERATOR ARCHITECTURE AND WORK FLOW

The last stage of analysis that is proposed to be implemented in the proposed System on Chip is the Convolutional Neural Network (CNN) accelerator [8]. When the biomedical signal has been filtered and converted the CNN is the one that interprets the data that has been processed and provides the pattern of the waves that will be classified as normal or abnormal physiological states [19] [15]. The real time and continuous monitoring is possible with the CNN implementation in hardware but the data privacy is also maintained as the analysis is done directly on the device, making it not required to process the information on an external server device. Table II summarizes the architecture of the implemented CNN model with convolution, pooling, and fully connected layers employed in the model classification [25] [3].

TABLE II : CNN ARCHITECTURE

Layer	Filters	Kernel	Output
Conv1	3	3	30x3
Pool1	—	2	15x3
Conv2	2	3	13x2
Pool2	—	2	7x2
FC	—	—	8 classes

### A. Relation with FIR and FFT

CNN accelerator is based on the results of the FIR filter and FFT processing units, and is used to accurately detect [9]. FIR filter then eliminates any undesired noise and interference thus only preserving the significant portions of the

biomedical waveform. This signal in the clean time-domain can be subsequently provided directly to the CNN or can be forwarded through the FFT step to extract frequency-domain features [11]. The FFT output is useful in pointing out periodic and spectral properties in the signal that are not easily noticed in the raw waveform which offers a second description of the signal. The combination of both areas of information gives a more discriminative and richer feature set to CNN [20] [8]. This enhances its capacity to differentiate normal physiological behavior and abnormal or disease-related behavior pattern. Therefore, FIR phase guarantees the signal quality, the FFT stage improves the clarity of features, and the CNN phase makes the last intelligent classification step - a sequence of intelligent diagnostic work within the System on Chip [16] [9].

#### B. Interaction between CNN and General Processor Core

CNN accelerator works in coordination with the RISC V processor. It is the processor that handles the input and gets it ready, sorts it into the appropriate sets of features and sends the control commands to the CNN hardware [23] [5]. When the CNN accelerator is engaged, convolution, activation, and pooling processes are computed with great computational intensity in hardware, instead of the processor, bringing considerable workload relief to the processor [16] [15]. Once inference is done, the CNN sends the classification results or confidence scores back to the processor which subsequently determines whether an alarm, logging entry, or follow up action is necessary [11]. Such collaborative labor of work makes it possible to conduct real time diagnostic analysis without losing its efficiency in the normal functioning of the system [16] [7].

#### VII. OVERALL SOC INTEGRATION

The RISC V processor of the proposed System on Chip acts as the central controller and it coordinates the work of the FIR filter, the FFT unit and CNN accelerator to create an autonomous biomedical processing pipeline. The processor then sets the FIR filter and makes it operational such that the noise in real-time biomedical signals is eliminated. The filtered output is then shared with the FFT module to analyze the frequencies or passed as-is to the CNN accelerator on the need to use the power of frequencies or not, respectively [22]. Both the FFT and CNN blocks are hardware coprocessors, so any computationally intensive task can be executed, and the main processor is not loaded [19] [9]. When the CNN finishes the process of classification the ultimate decision output is sent back to the processor to be logged or generate alerts. This ensures that the system is fast, has low power consumption, and can be successfully diagnosed on the device itself (no external desired computation required) by integrating all stages in a single SoC [19] [5].

#### VIII. RESULT AND DISCUSSION

The proposed System on Chip was tested through the analysis of the results of the processor, FIR filter and CNN accelerator. This was verified by the execution using the extra instruction set as the custom RISC V processor functioned properly and pipelined execution was verified. The FIR filter output revealed effective noise reduction without removing the relevant biomedical waveform characteristics and this indicates that hardware-based filtering is suitable in real time monitoring. A sample detection report generated by the CNN is shown in Fig. 5, where the final classification result is

displayed after signal analysis. In addition, the processor design was synthesized and verified at the hardware level, as illustrated in Fig. 6, confirming that the architecture is suitable for implementation on silicon or FPGA platforms. These combined results demonstrate that the proposed SoC architecture enables fast, secure, and efficient biomedical signal analysis in real time. The CNN accelerator quickly categorized signals processed into both normal and abnormal form without any external computation, and it gave quick onlap results [7]. All these results suggest that the integrated architecture facilitates real time, secure and efficient biomedical signal analysis.

=====	
ECG CNN ANALYSIS REPORT	
=====	
<b>Detection Results:</b>	
Normal ECG	: DETECTED
PVC	: NO
Tachycardia	: DETECTED
Bradycardia	: DETECTED
ST Elevation	: DETECTED
ST Depression	: DETECTED
High R-Wave	: DETECTED
Low R-Wave	: DETECTED
<b>Final Diagnosis</b>	<b>: NORMAL ECG</b>
=====	

Fig. 5. CNN Detection Result

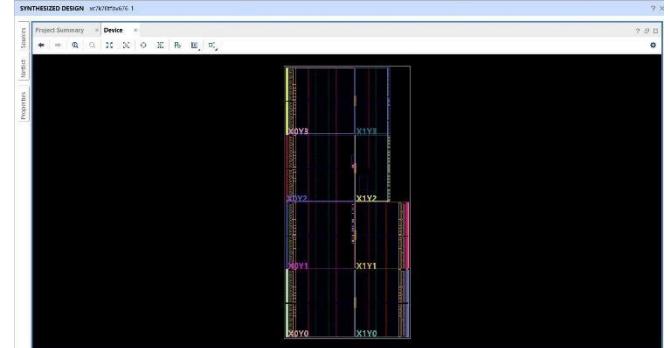


Fig. 6. Processor Synthesis Result

#### CONCLUSION

This paper discussed a heterogeneous RISC V based System on Chip that was developed with the specific purpose of real time biomedical signal analysis. Its architecture combines a common-purpose processor core with hardware processing units on FIR filtering, FFT processing, and CNN based [24] [16] classification, enabling noise removal, feature extraction and automated diagnosis to be run directly on the component. With the addition of both custom instructions and hardware acceleration, the system can have a higher processing speed and lower latency than conventional processor-based solutions, and provide patient data privacy by preventing data sent to a processing unit. The findings prove that the customization of an open processor platform on application specific hardware is an effective and efficient solution towards on-going biomedical monitoring and intelligent healthcare device solutions.

## REFERENCES

- [1] J. He et al., "Design and Implementation of a RISC-V SoC for Real-Time Epilepsy Detection," in Proc. IEEE Int. Conf. (SSD), 2023.
- [2] R. Rimolo-Donadio et al., "A RISC-V Based Medical Implantable SoC for High Voltage and Current Tissue Stimulus," in Proc. IEEE Latin American Symp. Circuits and Systems (LASCAS), 2020.
- [3] L. Guo et al., "68-Channel Neural Signal Processing System-on-Chip with Ultra-Low Power RISC-V Core and Hardware Accelerators," IEEE, 2024.
- [4] A. Gamatié et al., "Towards Energy-Efficient Heterogeneous Multicore Architectures for Embedded Systems," IEEE Trans. Comput., vol. 68, no. 1, 2019.
- [5] E. Choi et al., "Development of an Ultra-Low Power RISC-V Processor for Wearable Anomaly Detection," J. Syst. Archit., vol. 154, 2024.
- [6] G. Leone et al., "SYNtzulu: A Tiny RISC-V-Controlled SNN Processor for Near-Sensor Data Processing," in Proc. IEEE Int. Symp. Circuits and Systems (ISCAS), 2024.
- [7] M. S. Islam et al., "A Low-Cost Hardware Architecture of Convolutional Neural Network for ECG Signal Classification," IEEE Access, vol. 9, 2021.
- [8] K. Balasubramanian et al., "A Clinically-Grounded Survey Across EEG, ECG, and Other Biosignal CNN Hardware Platforms," IEEE, 2025.
- [9] K. M. Awan et al., "A Hardware-Efficient and Flexible Mini-InceptionNet Accelerator for 1-D CNN-Based ECG Analysis," in Proc. IEEE Conf., 2025.
- [10] S. Schönle, T. Burger, M. Arnold, and N. Felber, "A multi-sensor and parallel processing SoC for miniaturized medical instrumentation," *IEEE J. Solid-State Circuits*, vol. 53, no. 7, Jul. 2018.
- [11] A. González et al., "A 16-mm<sup>2</sup> 106.1-GOPS/W Heterogeneous RISC-V Multi-Core Multi-Accelerator SoC," in Proc. Eur. Solid-State Circuits Conf. (ESSCIRC), 2021.
- [12] J. Zuckerman et al., "Enabling Heterogeneous, Multicore SoC Research With RISC-V and ESP," in Proc. Workshop on Computer Architecture Research with RISC-V (CARRV), 2022.
- [13] A. Kurth et al., "HERO: Heterogeneous Embedded Research Platform for Exploring RISC-V Manycore Accelerators," in Proc. CARRV, 2017.
- [14] F. Conti et al., "A Heterogeneous RISC-V-Based SoC for Secure and Energy-Efficient IoT End-Nodes," in Proc. Int. Conf. Computer Design (ICCD), 2020.
- [15] S. Chen et al., "A Comprehensive Review of Hardware Acceleration Techniques for CNN-Based EEG Signal Processing," Electronics, vol. 13, no. 4, 2024.
- [16] V. Rawal et al., "Hardware Implementation of 1-D CNN Architecture for ECG Signal Classification," Biomed. Signal Process. Control, vol. 86, 2023.
- [17] A. Al-Harbi et al., "Efficient CNN Architecture on FPGA Using High-Level Modules for ExG Signal Classification," Microprocessors Microsyst., vol. 90, 2022.
- [18] Zhang et al., "Design of a CNN Accelerator for Multitask EEG Signal Classification," IEEE Trans. Circuits Syst. I, vol. 71, no. 6, 2024.
- [19] H. Li et al., "Hardware Implementation of CNN Based on FPGA for EEG Signal Classification," IEEE Access, vol. 9, 2021.
- [20] A. Ait El Mouden et al., "FPGA Implementation of a CNN Application for ECG Classification," Int. J. Heat Technol., vol. 39, no. 1, 2021.
- [21] Y. Liu et al., "A High-Performance Heterogeneous Hardware Architecture for Brain-Computer Interface Based on EEGNet," Sensors, vol. 24, no. 10, 2024.
- [22] D. Samakovlis et al., "A Benchmark Suite of TinyML Biomedical Applications for Real-Time Monitoring on Wearable Devices," in Proc. MLSys Workshop, 2024.
- [23] C. Koenig et al., "HeroSDK: Streamlining Heterogeneous RISC-V SoC Prototyping and Evaluation," ACM SIGARCH Comput. Archit. News, vol. 52, no. 2, 2024.
- [24] R. Torfah et al., "An Open-Source RISC-V Platform for Safety-Critical and Medical Applications," in Proc. Int. Conf. Hardware/Software Codesign and System Synthesis (CODES+ISSS), 2022.
- [25] M. Alioto et al., "Energy-Efficient Embedded Processing Systems for Wearable and Implantable Biomedical Devices," IEEE Trans. Circuits Syst. I, vol. 67, no. 10, 2020.