Heterogeneous Multicore SoC Architecture for Real-Time Biomedical Data Analysis

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

Subulakshmi A
Electronics and Communication Engineering
Chennai Institute of Technology
Chennai, Tamil Nadu, India
subulakshmia.citchennai.net

Abstract—**This paper describes the design of a heterogeneous multi core system on chip developed for real time biomedical data analysis. Many existing healthcare monitoring systems depend on cloud processing or general-purpose processors, which can lead to delays, higher power consumption, and concerns about data privacy. In this work, a custom RISC V based architecture is proposed that brings the processing capability directly onto the chip. The system includes a general-purpose core for control functions and a second processing core that is optimized for digital signal processing tasks. Custom hardware instructions are introduced to speed up filtering and frequency analysis operations that are commonly used in biomedical signal processing. A hardware convolutional neural network accelerator is also included to support automatic diagnosis on the device itself. Because all computation takes place locally, the system can respond faster and avoids exposing sensitive medical data to external networks. The proposed design shows that customized hardware built on an open processor platform can provide an efficient and secure solution for continuous biomedical monitoring applications.**

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

I. INTRODUCTION

Biomedical monitoring systems are becoming increasingly important in modern healthcare, especially with the growing need for continuous and real time patient observation. Signals such as the electrocardiogram carry valuable clinical information, but they are often affected by noise and must be processed quickly and accurately before any diagnosis can be made. Many existing systems rely on cloud platforms or general-purpose processors to handle this computation. While these approaches are convenient, they can introduce processing delays, require higher power consumption, and raise concerns about the privacy of sensitive medical information. These limitations make it difficult to achieve dependable real time performance in portable or embedded healthcare devices. To overcome these challenges, there is a growing interest in designing dedicated hardware platforms that are optimized specifically for biomedical signal analysis. Such systems can integrate computation directly within the device, reducing latency and ensuring that patient data remains local. In this work, a custom system on chip architecture based on the RISC V instruction set is developed for this purpose. The design uses a heterogeneous multi core approach, where one core performs general control functions and another core is tailored for digital signal processing operations. Additional hardware support is introduced to accelerate tasks such as filtering, frequency analysis, and automated pattern recognition. The aim of this architecture is to provide a secure, efficient, and scalable platform for continuous biomedical monitoring. By combining open processor technology with application specific hardware customization, the proposed system demonstrates a practical pathway toward faster and more reliable medical signal analysis in real time environments.

II. SYSTEM ARCHITECTURE AND COMPONENTS

The proposed System on Chip integrates several key hardware blocks that work together to support real time biomedical signal analysis. One of the core elements is the Finite Impulse Response (FIR) filter module, which is used to remove noise and unwanted disturbances from raw biomedical signals such as ECG waveforms. Since these signals are easily affected by motion, electrical interference, and environmental noise, reliable filtering is essential to ensure that meaningful diagnostic information is preserved. Implementing the FIR filter in hardware allows the system to clean the incoming data stream with minimal delay and lower power consumption than a conventional software approach. In addition to filtering, the system includes a dedicated Fast Fourier Transform (FFT) processing unit. This module converts the signal from the time domain into the frequency domain, making it possible to study the spectral content of the waveform and identify hidden frequency components or recurring noise patterns. Frequency analysis is particularly important in biomedical monitoring, as it helps in detecting abnormalities and preparing the data for further machine learning-based interpretation. By carrying out FFT operations directly in hardware, the SoC is able to achieve fast and efficient computation suitable for continuous real time processing.

To further enhance performance, the processor architecture incorporates custom extensions to the standard instruction set. These instructions are designed specifically for computationally intensive digital signal processing tasks such as multiply–accumulate operations and data rearrangement. Instead of executing these functions through multiple software instructions, the customized hardware instructions allow them to be completed in a single cycle. This reduces execution time, lowers energy consumption, and demonstrates the benefit of tailoring processor architecture to a target application such as biomedical analytics. Finally, the system integrates a hardware-based Convolutional Neural Network (CNN) accelerator for automatic diagnosis and pattern recognition. After the signal has been filtered and analyzed, the CNN processes the data to detect potential irregularities that may be linked to medical conditions. Running neural network inference within the SoC avoids the need to transmit sensitive information to external servers, improving both speed and data privacy. Together, these components create a secure, efficient, and highly responsive platform for real time biomedical monitoring.

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

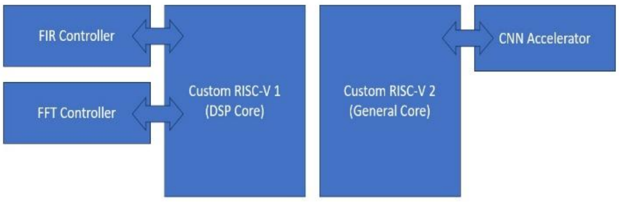


Fig. 1. System Architecture

III. CUSTOM RISC-V INSTRUCTION SET ARCHITECTURE

The System on Chip is developed using the RISC V instruction set architecture, which provides the flexibility to introduce new instructions tailored for specific applications. In this design, the base RV32I pipeline is extended with additional hardware logic to support custom instructions required for biomedical signal processing. The processor follows a pipelined execution flow, where each instruction passes through fetch, decode, execute, memory, and write-back stages. This structure improves throughput by allowing multiple instructions to be processed concurrently. The custom instructions are integrated into the decode and execute stages so that specialized digital signal processing operations can be executed directly in hardware without disrupting the standard instruction format. As a result, the processor maintains compatibility with conventional RISC V software tools while still achieving application-specific acceleration.

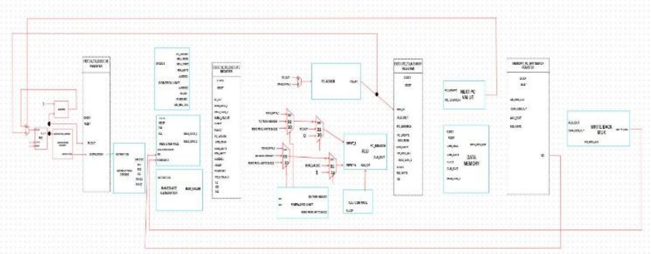


Fig. 2. Custom RISC V Architecture

A. VMAC Instruction for FIR Filter

One of the key custom instructions implemented in the processor is the Vector Multiply–Accumulate (VMAC) instruction. This instruction is designed to accelerate Finite Impulse Response filtering, which is widely used for noise removal in biomedical signals. In a conventional processor, FIR filtering requires repeated multiply and addition operations across multiple instructions, leading to increased latency and power consumption. The VMAC instruction performs these operations in a single execution cycle by multiplying the input samples with the corresponding filter coefficients and accumulating the result in hardware. This significantly reduces the total number of clock cycles required for filtering and allows the system to process biomedical signals in real time with improved efficiency.

B. Bit Reversal Instruction for FFT

Another important extension is the Bit-Reversal instruction that supports Fast Fourier Transform computation. FFT algorithms rely on bit-reversal addressing to reorder input data before frequency-domain processing. When implemented purely in software, this data rearrangement requires multiple memory and arithmetic operations, which increases processing overhead. By introducing a dedicated Bit-Reversal instruction, the processor is able to generate reordered indices directly through hardware logic, completing the task in a much shorter time. This accelerates the FFT pipeline and improves the responsiveness of frequency-based biomedical analysis, making the architecture suitable for continuous monitoring and rapid diagnostic applications.

Together, these custom instructions demonstrate how specialized hardware extensions within an open RISC V framework can provide substantial performance gains while retaining architectural simplicity and compatibility.

IV. FIR FILTER ARCHITECTURE AND WORKFLOW

The Finite Impulse Response (FIR) filter is a key component of the proposed System on Chip, as it performs noise reduction on biomedical signals before further processing takes place. Signals such as the electrocardiogram are often affected by motion artifacts, electrical interference, and background disturbances that can mask clinically important waveform details. The FIR filter is implemented directly in hardware so that this cleaning process happens continuously and without delay, ensuring that the signal remains suitable for real time medical monitoring.

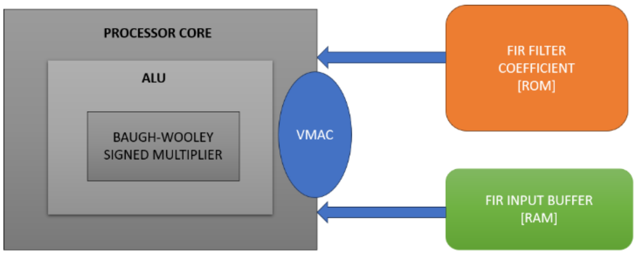


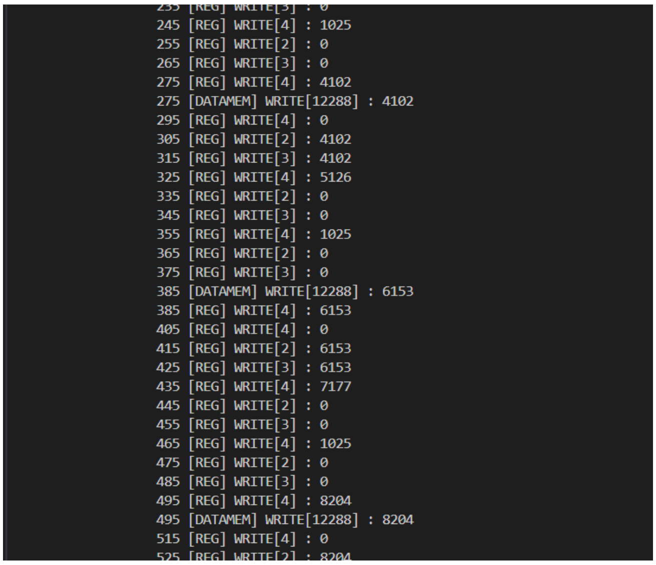
Fig. 3. FIR Filter Architecture

A. Functional Role and Data Flow

The main purpose of the FIR filter is to suppress unwanted frequency components while preserving the useful diagnostic information in the biomedical signal. During operation, each new sensor sample is stored in a buffer that holds a window of recent data points. These stored values are multiplied by predetermined filter coefficients, and the results are summed to generate the filtered output. This process repeats for every incoming sample, meaning that the filter works in a fully streaming manner. Because the computation is carried out in hardware, filtering is completed quickly with minimal latency compared to a software-based implementation.

B. *Instruction Level Acceleration using VMAC*

To further increase processing efficiency, the multiply–accumulate operations inside the FIR filter are supported by the custom Vector Multiply Accumulate instruction integrated into the RISC V processor. Instead of executing multiple instructions for multiplication and addition, the VMAC instruction performs the operation in a single hardware cycle. This reduces execution time, lowers power consumption, and allows the FIR filter to operate continuously without interrupting real time data flow. As a result, a clean and stable biomedical signal is delivered to the subsequent FFT and CNN processing stages, improving overall diagnostic accuracy and system performance.



```
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

The Fast Fourier Transform (FFT) module forms another major processing stage in the proposed System on Chip. While the FIR filter cleans the raw biomedical waveform in the time domain, the FFT converts the filtered signal into the frequency domain so that hidden spectral components can be identified. This is particularly useful in biomedical monitoring, as certain abnormalities and recurring disturbances are easier to detect through frequency analysis than from the raw waveform alone. Implementing the FFT in hardware ensures that this computation can be carried out continuously without interrupting real time system operation.

A. *Functional Role and Data Flow*

During operation, blocks of filtered signal samples are collected and passed into the FFT processing unit. The algorithm then decomposes the signal into its discrete frequency components, allowing the system to observe periodic patterns, noise behavior, and spectral features relevant to diagnosis. This frequency-domain information may also be used as an input to the CNN accelerator for classification. Since this computation is handled by a dedicated hardware block, the overall latency is greatly reduced compared to performing FFT entirely in software. This allows the system to support continuous and responsive biomedical monitoring.

B. *Instruction Level Acceleration using VMAC*

To further enhance FFT performance, the processor incorporates a custom Bit Reversal instruction that assists with the data reordering required by the FFT algorithm. In conventional implementations, this step involves multiple memory accesses and arithmetic operations, which increases processing time. With the dedicated instruction, index reordering is handled directly in hardware, allowing the FFT engine to proceed more efficiently through its computation stages. This reduces the overall execution workload and ensures that frequency analysis keeps pace with the incoming biomedical signal stream. As a result, the system is able to deliver frequency-domain insights rapidly, supporting accurate and timely medical assessment.

VI. CNN ACCELERATOR ARCHITECTURE AND WORK FLOW

The Convolutional Neural Network (CNN) accelerator forms the final stage of analysis in the proposed System on Chip. Once the biomedical signal has been filtered and transformed, the CNN is responsible for interpreting the processed data and identifying patterns that may correspond to normal or abnormal physiological conditions. Integrating the CNN directly into hardware allows the system to support continuous real time monitoring while maintaining data privacy, since no external server processing is required.

TABLE II : CNN ARCHITECTURE

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

A. *Relation with FIR and FFT*

The CNN accelerator relies on the outputs of both the FIR filter and FFT processing stages to perform reliable detection. The FIR filter first removes unwanted noise and interference so that only the meaningful portions of the biomedical waveform are preserved. This clean time-domain signal may then either be supplied directly to the CNN or passed through the FFT stage for frequency-domain feature extraction. The FFT output highlights periodic and spectral characteristics that may not be obvious in the raw waveform, providing an additional representation of the signal. By combining information from both domains, the CNN receives a richer and more discriminative feature set. This improves its ability to distinguish between normal physiological behavior and abnormal or disease-related patterns. Thus, the FIR stage ensures signal clarity, the FFT stage enhances feature visibility, and the CNN stage performs the final intelligent classification — forming a complete and coordinated diagnostic workflow within the System on Chip.

B. *Interaction between CNN and General Processor Core*

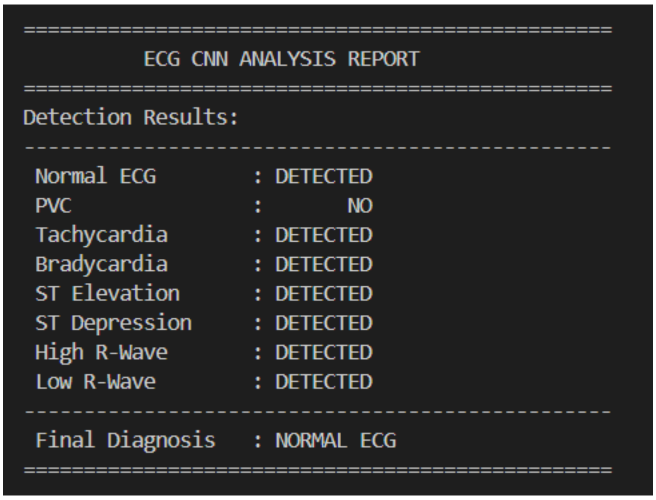
The CNN accelerator operates in close coordination with the RISC V processor. The processor is responsible for preparing the input data, organizing it into suitable feature sets, and issuing control commands to the CNN hardware. Once activated, the CNN accelerator performs the computationally intensive convolution, activation, and pooling operations directly in hardware, significantly reducing the workload on the processor. After inference is completed, the CNN returns classification outputs or confidence levels back to the processor, which then decides whether an alert, log entry, or follow-up operation is required. This cooperative workflow ensures that real time diagnostic analysis can be carried out without compromising the efficiency of normal system control functions.

VII. OVERALL SOC INTEGRATION

In the proposed System on Chip, the RISC V processor operates as the central controller that coordinates the FIR filter, FFT unit, and CNN accelerator to form a complete biomedical processing pipeline. The processor first configures and enables the FIR filter so that incoming biomedical signals are cleaned of noise in real time. The filtered output is then either forwarded to the FFT module for frequency-domain analysis or supplied directly to the CNN accelerator, depending on the required detection mode. Both the FFT and CNN blocks function as hardware co-processors, allowing computationally intensive operations to be carried out without burdening the main processor. Once the CNN completes classification, the final decision output is returned to the processor for logging or alert generation. By integrating all stages within a single SoC, the system achieves fast processing, low power consumption, and secure on-device diagnosis without the need for external computation.

VIII. RESULT AND DISCUSSION

The proposed System on Chip was evaluated by analyzing the outputs from the processor, FIR filter, and CNN accelerator. The custom RISC V processor operated correctly with the extended instruction set, confirming reliable pipelined execution. The FIR filter output showed effective noise reduction while preserving the key biomedical waveform features, demonstrating the suitability of hardware-based filtering for real time monitoring. The CNN accelerator successfully classified the processed signals into normal and abnormal patterns, providing fast on-device inference without external computation. These results collectively indicate that the integrated architecture supports efficient, secure, and real time biomedical signal analysis



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

Fig. 5. CNN Detection Result

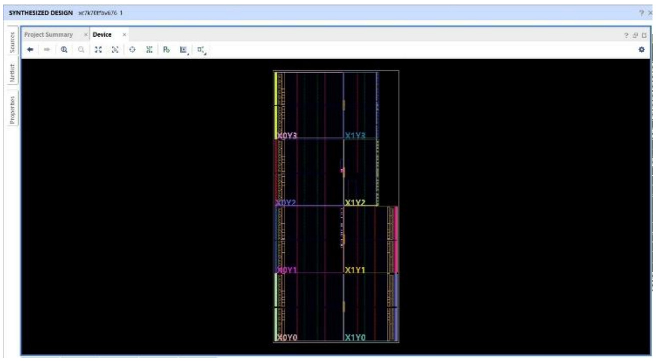


Fig. 6. Processor Synthesis Result

CONCLUSION

This work presented a heterogeneous RISC V based System on Chip designed specifically for real time biomedical signal analysis. The architecture integrates a general-purpose processor core with dedicated hardware modules for FIR filtering, FFT processing, and CNN based classification, allowing noise removal, feature extraction, and automated diagnosis to be carried out directly on the device. By introducing custom instructions and hardware acceleration, the system achieves higher processing speed and lower latency compared to conventional processor-based solutions, while also maintaining patient data privacy by avoiding external computation. The results demonstrate that application specific hardware customization on an open processor platform offers a practical and efficient approach for continuous biomedical monitoring and intelligent healthcare devices.

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.