



INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

RESOURCE-EFFICIENT VLSI ARCHITECTURE FOR ECG HEARTBEAT CLASSIFICATION USING FPGA-BASED HARDWARE–SOFTWARE CO-DESIGN

¹T. Rohith, ²Dr. Snehalatha

¹ Student, ² Associate Professor

¹VLSI System Design,

^{1,2}JB Institute of Engineering and technology, Hyderabad, India

Abstract: In this paper, a resource-efficient modeling framework for co-design of hardware and software is proposed to transmit ECG signal in real time and classify them in an automatic manner based on the embedded machine learning model. The raw ECG signal goes through pre-processing in MATLAB, is converted into a 16-bit signed hexadecimal 2's complement format and is written to a COE (memory initialization) file. This file loads the Block RAM (BRAM) blocks into the Xilinx Basys3 FPGA board using the Xilinx Vivado Block Memory Generator IP. The following custom made Verilog UART Transmitter module reads ECG samples sequentially from BRAM and transmits the sample values serially at 9600 baud rate. At the receiver end, the data received via UART is passed to a Raspberry Pi which reconstructs the signed 16-bit ECG samples using two's complement, classifies them as normal or abnormal, and passes the results to a trained Random Forest classifier. The designed architecture shows high reliability in transferring the data, moderate utilization of FPGA resources, minimum computational load on the embedded platform and it is scalable for implementing an intelligent real-time cardiac health monitoring pipeline.

Index Terms - ECG classification, FPGA, UART, Random Forest, hardware–software co-design, Raspberry Pi, BRAM, Verilog HDL, biomedical signal processing.

I. INTRODUCTION

Cardiovascular diseases (CVDs) continue to be one of the biggest killers around the world, claiming millions of lives every year. Heart-related ailments like arrhythmia, myocardial infarction, atrial fibrillation and sudden cardiac arrest are becoming very common, indicating an urgent need for constant and stable cardiac monitoring systems. Early detection of abnormal cardiac activity is extremely beneficial for the effectiveness of clinical treatments and decreases the mortality rate. Thus, designing smart, portable, real-time cardiac monitoring systems is an important research field for biomedical engineers and embedded healthcare technologies.

An electrocardiogram (ECG) measures the electrical activity of the heart using a non-invasive diagnostic technique and is the most popular test for doing this. An ECG wave can be very useful for information about the heart rhythm, the conduction pathways, the problems of the ventricles, the problems of the atria, and various cardiovascular problems. Typical segments of the tracing, These (P wave, QRS complex and T wave) are examined by doctors when assessing whether something has gone wrong, and determining treatment strategies. Due to its simplicity and reliability in detecting cardiopathy clinically, ECG is low cost, non-invasive and used widely to assess CVS clinically. It is used as a diagnostic tool in the hospitals, ambulatory monitoring devices and as a wearable health care device. Though this is common, clinicians' traditional interpretation of an ECG relies primarily on manual and clinical-grade acquisition systems.

interpretation by cardiologists. When 24-hour patient data are recorded, there is a vast amount of information to go through to avoid misdiagnosis. Manual analysis takes time to process and can be time-consuming, and can introduce inaccuracies, particularly when combined with differences between analysts and subject matter. subtle changes occur. Also, clinical analysis is centralized, which provides the challenges in providing immediate clinical feedback, and With so many issues that hinder remote medical practice and ongoing patient monitoring. This has increased the attractiveness of automation of ECG analysis. appealing interactive research focused on turning rapid and precise diagnosis of heart disease into an activity as little as possible human involvement possible.

The field of automated ECG classification has gone along with the latest advancements of machine learning and embedded computing, improving by leaps and bounds. Highly developed and now applicability on portable health care devices. The discriminative function can be learnt by machine learning features of ECG signals and, with high accuracy, classify normal and abnormal cardiac rhythms. .The Random Forest classifier, one A supervised learning technique, has received a lot of attention due to its strong classification accuracy and immunity to noisy data entries. biomedical signals, high resistance to overfitting and low computation complexity. The following properties make it an outstanding material for: Embedded systems that are real-time, with limited resources – memory and CPU – and energy.

To obtain the necessary real-time performance, high-speed signal acquisition, storage and communication demand efficient hardware platforms. Field Programmable Gate Arrays (FPGAs) are a suitable technology choice as they offer massive parallel processing, deterministic timing, hardware-level reliability, and low-latency data handling. FPGA architectures can perform many hardware operations concurrently, which makes them ideally suited for biomedical signal processing applications, unlike conventional processors. Efficient data buffering, pre-processing and high-speed serial communication with little processing delay are realized through configurable logic resources, Block Random Access Memory (BRAM) and communication peripherals. There has been a surge in FPGA usage in medical instrumentation, wearable monitoring systems and edge healthcare devices.

While FPGAs are well suited to deterministic signal processing, complex machine learning algorithms can require considerable hardware resources and longer design times. .In these days of embedded processors such as Raspberry Pi are available, which would provide an alternative Execute high-performance machine-learning models of small size, low cost and low power consumption. The machine learning runs on the Raspberry Pi carries the machine learning. Python libraries for rapid deployment of classification models without specialized support hardware accelerators. When deterministic data acquisition using FPGA is combined with intelligent classification using Raspberry Pi it yields an efficient Hardware-Software co-design platform to meet a balance between computation performance, flexibility and implementation cost. This paper suggests a comprehensive framework for the automated ECG classification of hardware-software designed system and uses the MATLAB software. FPGA and a raspberry pi. The ECG signals are initially preprocessed in MATLAB to remove noises, normalize the amplitude and packet them into signal windows. Acquire representative samples for its hardware implementation. The processed data are stored in the FPGA Block RAM (BRAM) for deterministic memory access. A UART module designed on the FPGA communicates with the Raspberry Pi in serial fashion, providing reliable, low-latency communication. The received ECG samples are analyzed using a Random Forest classifier on the Raspberry Pi to classify them as normal or abnormal based on learned decision boundaries. This architecture combines the reliability of FPGA hardware with the intelligence of machine learning software.

The proposed architecture presents several advantages compared to conventional FPGA-only or software-only ECG classification systems. The FPGA provides reliable real-time storage and transmission of signals via BRAM and UART interfaces, while the Raspberry Pi runs the computationally efficient machine learning algorithms without consuming extra FPGA resources. This hardware–software partitioning reduces system complexity, improves scalability, and enables future integration with IoT healthcare platforms, cloud-based medical services, and wearable patient monitoring devices. The modular design also enables easy upgrades, since incorporating more sophisticated classification algorithms does not require changes to the FPGA hardware design.

The main contributions of this research work are listed below:

- Design and implementation of an extensive and integrated automated ECG analysis system across the FPGA, MATLAB and the Raspberry Pi.
- Reliable serial communication via the UART instead of other communications methods, as well as efficient application of ECG sample memory using Block RAM on the FPGA, are achieved interface.
- Normal or Abnormal (Arrhythmia) stoned as a result of the accurate classification of the ECG signals using a machine learning classifier (Random Forest model) that is deployed on the Raspberry Pi real-time normal/abnormal. Processing and reliable and stable data delivery in the application. Reliable communication between the involved hardware, reliable and deterministic signal processing and data delivery in use Transmission, and the ability to recognise cardiac diseases appropriate for the hand-held medical application.

In this paper, the existing work for cardiac FPGA processing and machine learning based on FPGA are reviewed in Section II. classification. The overall system architecture, ECG pre-processing methodology and ECG signal processing on FPGA Description are introduced in the Section III. Implementation such as the BRAM storage module and UART communication module. Results of the experiments are provided in Section IV. Section In order to conclude the paper, directions for future research for intelligent embedded healthcare systems are presented.

1.1 Problem Statement

The immediate transmission of ECG signal, efficiently and without man power intervention, is a challenging task for current healthcare systems. The solutions that are currently available are either software based which causes a delay, or they are based on expensive medical grade hardware. The main issues tackled in this research are: (i) storing the ECG in FPGA memory, (ii) correct and accurate UART .Because of that, the following features of the transmission and reconstruction at the receiver must be achieved: (iii) reconstruction of signed samples at the receiver, (iv) integration of machine learning, and (v) synchronisation of hardware and software.

1.2. Contributions

The main merits of this book are:

- An end-to-end hardware–software co-design framework for ECG monitoring.
- FPGA based BRAM storage and UART data loss-less ECG Streaming.
- Two's complement decoding for accurately reconstructing a signed sample.
- With the Raspberry Pi you can embed your own Random Forest classification.

II. LITERATURE REVIEW

In recent decades, numerous studies have been conducted on electrocardiogram (ECG) signal analysis, which is essential for diagnosing cardiovascular diseases. These studies span classification algorithms involving machine learning, embedded systems design, and FPGA-based biomedical systems in addition to ECG signal processing.

The real-time QRS detection algorithm by Pan and Tompkins [1] is a widely used algorithm for extracting ECG features in real time and laid the groundwork for automated ECG feature extraction. Addison [2] showed that wavelet-based decomposition is an effective tool for removing noise and detecting morphological features from the ECG compared to traditional filtering techniques, improving signal quality before classification. Sriram and Eswaran [3] developed an FPGA-based ECG signal processor, showing that real-time QRS detection can be achieved with much lower latency using hardware than a microcontroller. Similarly, Zhang et al. [4] presented a hardware-efficient FPGA-based ECG monitoring system with low power requirements for portable healthcare applications performing real-time filtering and feature extraction.

Kumar's et al representation [5] of the feasibility of the use of low cost embedded Linux platforms represents an attempt to propose an embedded Linux platform for biomedical applications.

An Electronic Chart System which communicates with the cloud for remote monitoring. Osowski and Linh [6] used fuzzy hybrid neural .Their methods have resulted in high classification accuracy for ECG beats in networks, but consume a lot of computing power, making them impractical for this type of recognition. On resource constrained embedded systems. The Random Forest algorithm suggested by Breiman [7] is an ensemble learning model comprising of multiple decision trees and aggregates their decisions. It is not prone to overfitting, has simple calculations, and is highly interpretable, and it is resistant to noise — all the desirable properties in embedded classification problems. Using machine learning methods for automatic classification of surgical procedures. Machine learning techniques for automatic classification of surgical procedures validated by Acharya et al. [8]. A system has been developed to detect arrhythmia from ECG signals which achieves competitive classification performance on several cardiac conditions. In order to reduce the bandwidth and power consumption of the wireless biomedical communication, Spagnolo et al. [9] proposed a compressive .A sensing technique for data transmission of ECG signals securely and efficiently. Ali was worked to develop a reconfigurable UART IP core [10]. Designed for IoT applications demanding flexible and reliable serial communications for medical data streams, for FPGAs. Mohan et al. [11] enumerated different types of neural networks whose implementations have been done by FPGAs for Biomedical applications and presented the advantages and disadvantages of each. Sharma et al. [12] proposed a multi-functional UART system integrated with HyperTerminal, proving highly sustainable for embedded healthcare systems requiring robust serial communication. High accuracy in ECG heartbeat classification has been achieved using CNNs, though at greater computational cost than classical methods [13]. For signal detection and adaptive channel estimation in IoT networks, Kapileswar and Phani Kumar [14] proposed optimized deep learning techniques demonstrating efficient model deployment under representative IoT platform constraints. Hybrid intelligent architectures for real-time classification, such as the graph neural network and BiLSTM hybrid method of Simon and Kapileswar [15] for transactional fraud detection, are also increasingly common.

From the literature, there is a clear gap in FPGA-based ECG streaming combined with on-device Random Forest classification using BRAM within a single, end-to-end validated pipeline. Existing work largely focuses on either signal acquisition or classification, with these tasks most commonly performed separately, often in the cloud rather than on the embedded system itself. The proposed work addresses this gap by incorporating all these stages into a complete, hardware-verified co-design framework.

III. PROPOSED METHODOLOGY

The proposed system follows a hardware–software co-design architecture with five stages, illustrated in Fig. 1.

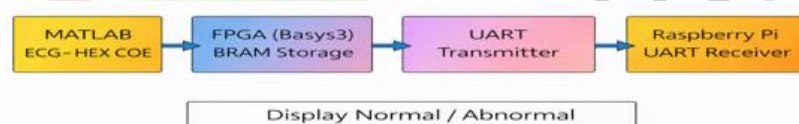


Figure .1. Block diagram of the FPGA-based ECG classification system

3.1 ECG Signal Conversion in MATLAB

The ECG signal is acquired from a standard benchmark .mat file. Raw signal readings are read and stored as 16-bit signed integers. To make the data compatible with FPGA BRAM initialization, the signal is converted to two's complement hexadecimal format and exported as a COE (Coefficient) file with memory_initialization_radix = 16. This stage assures consistency of data in the between This-Chase area. The MATLAB environment to the FPGA memory architecture.

3.2 FPGA BRAM Initialization

The COE file is created from this and uploaded into the Xilinx Vivado Block Memory Generator IP. The BRAM is 16-bit wide and has a the memory size is equal to the number of ECG samples recorded, and every memory address holds an ECG sample. The static ECG is recorded by BRAM signal: Deterministic and glitch-free access when transmitting through UART.

3.3 UART Transmitter Module Design

VHDL Transmitter Using a custom UART. The address counter is a sequential address counter that controls the BRAM reads

To accept synchronous BRAM latency, inserted 2 wait cycles. All valid 16 bit ECG samples are recorded and divided into a high byte Each sent serially as either as a (MSB) or low-byte (LSB). UART is set to 9600 to be interoperable with the Raspberry Pi baud, one start bit, one stop bit with 8N framing.

3.4 Raspberry Pi Signal Reception and Reconstruction

Raspberry pi configures itself as receiver and classifier. Using the Python serial library, the UART port is opened over 9600 baud.

The byte pairs detected are collected to make into 16-bit values and are converted back to ECG samples by decoding in two's complement. The We will store the reconstructed waveform as a NumPy array, which can be manipulated for feature extraction and classification.

3.5 Random Forest Classification

To address the above characteristics, the Random Forest model was chosen for the sound of hard, low cost and easy to grasp, it is suitable for the decision-making model of the new smart home embedded hardware. The model is trained off-line, and the trained model and the serialized data set are stored in the "joblib" format. The Raspberry Pi Passes reconstructed ECG data to trained model to generate a new ECGRs that are both statistic and morphology features bipolar output: Normal or Abnormal (Arrhythmia).

TABLE I. HARDWARE–SOFTWARE TASK PARTITIONING

FPGA (Hardware)	Raspberry Pi (Software)
BRAM Storage	Signal Reconstruction
UART Transmission	ML Classification
Deterministic Timing	Intelligent Decision

The selected equipment is the Basys3 FPGA board (Artix-7) and the Raspberry Pi which is used as the embedded station board. Signals are preprocessed in MATLAB (MathWorks), the design of the FPGA is done in Vivado (Xilinx), and The Python version of embedded classification. Python 2 version of embedded classification. Data communication was realized by UART techniques with the baud rate of 9600 baud, Machine learning used was Random Forest. The ECG signal set was pre-processed in MATLAB and then loaded onto the FPGA as a To use the Vivado Block Memory Generator IP allocate 16-bit hexadecimal (16 hex) COE file to program the Block RAM (BRAM). The UART The ECG image was then transmitted sequentially by the transmitter module to the Raspberry Pi processor in order to reconstruct the ECG into an image and learn to classify it. It was used to train a Random Forest model and then apply it for predictions.

3.6 Signal Transmission Results

All test samples were correctly processed in the UART transmission module implemented on the FPGA. The spatial resolution of each sample is 16 bits. Sent serially @ 9600 baud, over two 8-bit frames - MSB first, then LSB. No loss or syncing data

Problems arose transmitting the data and they were all accurately reconstituted on the Raspberry Pi using the two's complement method. conversion. The result achieved on the reconstructed waveform from Raspberry Pi shows the same behavior as the original MATLAB waveform, which proves the correct behavior of the BRAM initialization and UART communication integrity. The effective inoculation rate is represented by:

$$9600 \text{ bits/sec} \div 16 \text{ bits/sample} \approx 600 \text{ samples/sec}$$

This is approximately the standard ECG sampling rate of 250-500 Hz and this indicates a real-time streaming feasibility, the proposed architecture.

3.7 Classification Results

The Random Forest classifier was trained offline using labeled ECG benchmark data, and tested for inference on the Raspberry Pi. The Raspberry Pi was used to receive and reconstruct the ECG samples sent by the FPGA, and remove all other unnecessary data. Extracted features from the reconstructed signals and obtained the binary output ('Normal' or 'Abnormal') for the trained model. The terminal on the Raspberry Pi was used to show the classification result. For all the ECGs that he tested, the system was able to detect and accurately classified the signals. Although it is not as accurate as deep learning model like CNN, the RF classifier is much less complicated, demanding and requires less: It is suitable because it's understood decision output, requires less memory, is faster to infer and has computational power. To resource constrained embedded systems.

3.8 System Validation

The whole pipeline was validated: MATLAB simulated ECG signals stored in FPGA memory, sent to FPGA through transmitter and received by the receiver chip. The UART data from the FPGA to the Raspberry Pi which is reconstructed from the received samples is compared to the original UART data produced by MATLAB. The classification results were compared with the ground-truth labels and the classification was provided using the waveform. There were no errors anywhere along the pipeline, Ensures proper hardware and software component integration.

3.9. Resource Utilization and System Efficiency

The FPGA design was completely restricted to the logic needed to solve the problem – a Block RAM for storing the ECG data and simple combinational blocks for computing the results logic for the UART transmitter. There's no large heavy duty DSP computation on the FPGA which makes the hardware design simple and power efficient. Task partitioning between hardware and software is ideal – the FPGA used for deterministic, time-critical data storage, means the software is free of real-time computations and parallel operations. The Raspberry Pi processes machine learning inference for computations where flexibility is required, while the device with its strong memory capability processes the data in transport. The low latency which arises at the end of this co-design. A reliable classification, good signal transmission and reconstruction, and a modular, scalable architecture which can be extended with additional modules for multi-lead ECG monitoring or wireless communication.

IV. RESULTS AND DISCUSSION

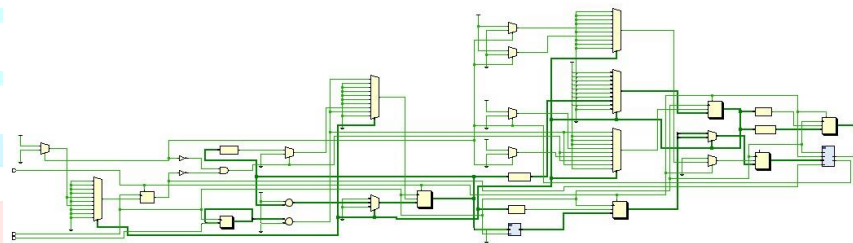


Figure.2.A UART transmitter module is shown in the following schematic which is drawn in RTL.

The UART transmitter module created in Vivado is presented in the RTL schematic in Fig. 2. The timing simulation The results show the correct synthesis of the serial communication logic on FPGA Board based on Artix-7 FPGA using minimal FPGA resources utilization.

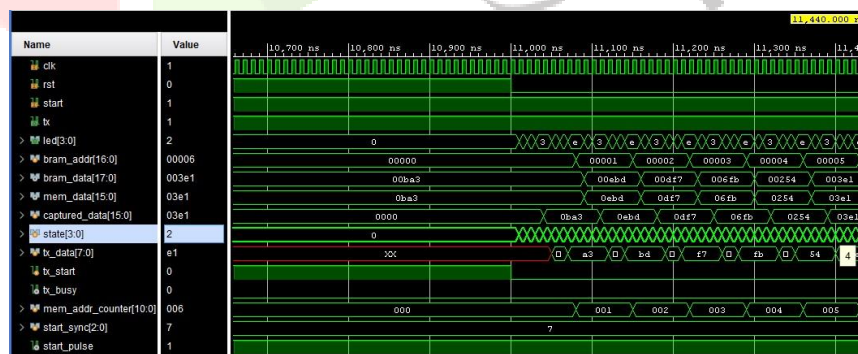


Figure.3. Simulation waveform for UART transmission

The behavior of the UART transmission observed in the Vivado simulation is correct as depicted in Fig. 3 above. Each 16 bit ECG reading is divided. These are transferred to MSB and LSB bytes and sent one after another. Simulations were performed to determine whether any data corruption would occur, and showed it to be safe. Read logic into state machine and BRAM.

Name	Slice LUTs (20800)	Slice Registers (41600)	Slice (815 0)	LUT as Logic (20800)	LUT Flip Flop Pairs (20800)	Block RAM Tile (50)	Bonded IOB (106)	BUFGCTRL (32)
uart_transmitter	70	78	30	70	47	2	8	3
design_inst (design_1_...	0	0	0	0	0	2	0	2
uart_tx_inst (uart_tx)	35	31	17	35	20	0	0	0

Figure.4. .Calculate the area utilization of FPGA.

The report of area utilization of FPGA that is achieved using Vivado is depicted in Fig. 4. The design has very little and is very efficient in terms of hardware. Low resource utilization: Small amount of Block RAM used for storage, simple part of sequential logic used for UART, indicating low. Low resource FPGA device-compatible hardware overhead.

Name	Slack	Levels	Routes	From	High ...	To	Total ...	Logic Delay	Net ...	Logic %	Net %
Constrained Paths (100)											
Path 1	4.004	0	1	mem...JC	4	desi...R[8]	1.658	0.518	1.140	31.2	68.8
Path 2	4.040	0	1	mem...JC	4	desi...[13]	1.622	0.518	1.104	31.9	68.1
Path 3	4.048	0	1	mem...JC	5	desi...R[7]	1.614	0.518	1.096	32.1	67.9
Path 4	4.051	0	1	mem...JC	6	desi...[11]	1.612	0.518	1.094	32.1	67.9
Path 5	4.096	0	1	mem...JC	8	desi...R[9]	1.566	0.456	1.110	29.1	70.9
Path 6	4.122	0	1	mem...JC	6	desi...R[6]	1.541	0.518	1.023	33.6	66.4
Path 7	4.169	0	1	mem...JC	5	desi...[12]	1.493	0.518	0.975	34.7	65.3
Path 8	4.176	0	1	mem...JC	8	desi...R[4]	1.487	0.456	1.031	30.7	69.3
Path 9	4.187	0	1	mem...JC	7	desi...R[5]	1.476	0.518	0.958	35.1	64.9
Path 10	4.201	0	1	mem...JC	9	desi...R[3]	1.461	0.456	1.005	31.2	68.8
Path 11	4.239	0	1	mem...JC	7	desi...[10]	1.423	0.456	0.967	32.0	68.0
Path 12	4.340	0	1	mem...JC	4	desi...R[8]	1.320	0.518	0.802	39.2	60.8
Path 13	4.376	0	1	mem...JC	4	desi...[13]	1.284	0.518	0.766	40.3	59.7
Path 14	4.384	0	1	mem...JC	5	desi...R[7]	1.276	0.518	0.758	40.6	59.4
Path 15	4.387	0	1	mem...JC	6	desi...[11]	1.274	0.518	0.756	40.7	59.3
Path 16	4.432	0	1	mem...JC	8	desi...R[9]	1.228	0.456	0.772	37.1	62.9
Path 17	4.458	0	1	mem...JC	6	desi...R[6]	1.203	0.518	0.685	43.1	56.9
Path 18	4.505	0	1	mem...JC	5	desi...[12]	1.155	0.518	0.637	44.8	55.2

Figure.5. This report is a timing analysis (delay) of the material.

The timing analysis report generated by Vivado is illustrated in Fig.5. There were no violations of the setup and hold requirements; all timing requirements are met – has the design passed this test and is performing as it's expected to at the clock speed used.

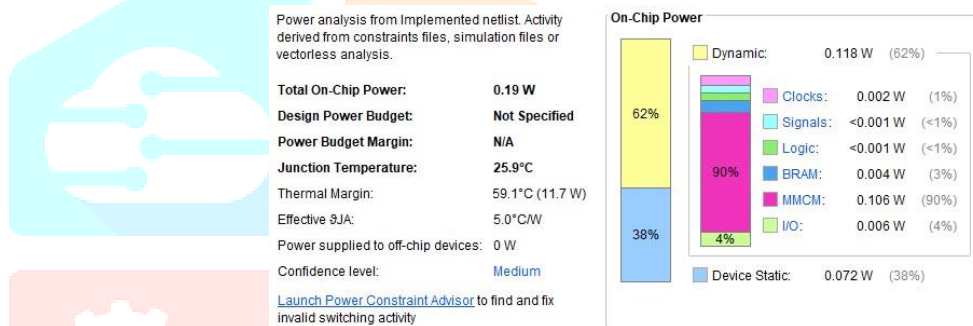


Figure.6. A report of the Power Consumption of the FPGA

The power analysis data obtained from Vivado is shown in Fig. 6. Since the FPGA design only requires BRAM and UART logic and no logic blocks, this type of design is memory intensive. If there is no significant DSP computation, its dynamic power consumption is minimal and total hardware power consumption is decreased.

V. CONCLUSION

This paper offered a hardware–software co-design methods for real-time transmitting and classification of ECG signal. Digitized

Data from the ECG sensor was pre-processed and recorded within a COE initialized and stored in FPGA Block RAM and sent to a Raspberry Pi via a bus. A custom made Verilog UART module. After reconstruction on Raspberry Pi the signal was classified using a Random Forest model. The End-to-end pipeline with no synchronization errors and data loss. The resulting design was confirmed both to be correct and efficient on the Artix-7 FPGA through RTL synthesis, simulation, and post-implementation analysis. The modular and extensible design is proposed to be extended to real-time ECG sensor acquisition, wireless transmission and multi-lead monitoring. Integration with IoT cloud platforms and evaluation of deep-learning classifiers on more powerful embedded devices will be explored in future work.

REFERENCES

- [1] J. Pan and W. J. Tompkins, "A real-time QRS detection algorithm," *IEEE Trans. Biomed. Eng.*, vol. BME-32, no. 3, pp. 230–236, Mar. 1985.
- [2] P. S. Addison, "Wavelet transforms and the ECG: a review," *Physiol. Meas.*, vol. 26, no. 5, pp. R155–R199, 2005.
- [3] T. Sriram and C. Eswaran, "FPGA-based real-time ECG signal processor for QRS detection," in *Proc. IEEE Int. Conf. Electron. Des.*, 2012, pp. 1–6.
- [4] Y. Zhang et al., "Hardware-efficient ECG monitoring system using FPGA," *J. Med. Syst.*, vol. 39, no. 9, p. 90, 2015.
- [5] R. Kumar et al., "Raspberry Pi-based ECG monitoring system with cloud integration," in *Proc. IEEE ICECA*, 2018, pp. 1–5.
- [6] S. Osowski and T. H. Linh, "ECG beat recognition using fuzzy hybrid neural network," *IEEE Trans. Biomed. Eng.*, vol. 48, no. 11, pp. 1265–1271, 2001.
- [7] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
- [8] U. R. Acharya et al., "Deep convolutional neural network for the automated detection and diagnosis of seizure using EEG signals," *Comput. Biol. Med.*, vol. 100, pp. 270–278, 2018.
- [9] F. Spagnolo, B. Lal, P. Corsonello, and R. Gravina, "A novel compressive sensing method for secure and energy-efficient ECG signal transmission," *IEEE J. Biomed. Health Inform.*, vol. 29, no. 5, pp. 3439–3450, May 2025.
- [10] M. A. Ali, M. A. Pasha, S. Masud, and F. A. Ghazali, "FPGA design of a reconfigurable UART IP core and its SoC implementation for IoT applications," in *Proc. INMIC*, 2023, pp. 1–6.
- [11] N. Mohan, A. Hosni, and M. Atef, "Neural network implementations on FPGA for biomedical applications: a review," *SN Comput. Sci.*, vol. 5, p. 1004, 2024.
- [12] N. Sharma et al., "Design and implementation of a sustainable FPGA-based UART with HyperTerminal integration," in *ICSD 2023, CCIS*, vol. 1939, Springer, 2023, pp. 1–10.
- [13] M. Chourasia, A. Thakur, S. Gupta, and A. Singh, "ECG heartbeat classification using CNN," in *Proc. IEEE UPCON*, 2020, pp. 1–6.
- [14] N. Kapileswar and P. Phani Kumar, "Optimized deep learning for signal detection and adaptive channel estimation in underwater acoustic IoT networks," *Int. J. Commun. Syst.*, vol. 37, no. 4, p. e5673, 2024.
- [15] J. Simon and N. Kapileswar, "A graph neural and BiLSTM hybrid network for transactional fraud detection in digital payment gateways," in *Proc. CMSS 2025*, pp. 752–757, IEEE.