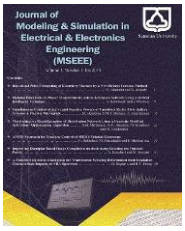
Semnan University

Journal of Modeling & Simulation in Electrical & Electronics Engineering (MSEEE)

Journal homepage: **https://mseee.semnan.ac.ir/**

ISSN: 2821-0786



Low-Power FPGA-Based Quantized Neural Network Accelerator for Real-Time Arrhythmia Detection from Single-Lead ECG

Amir Hossein Rahmani[1]*

***Abstract*--Arrhythmia detection from long-term electrocardiogram (ECG) recordings is essential for early diagnosis of cardiovascular diseases in portable and wearable monitoring systems. This paper presents a low-power hardware accelerator for real-time beat-level arrhythmia classification using an 8-bit quantized 1D convolutional neural network (QNN) implemented on a Xilinx Artix-7 FPGA. The network is trained and evaluated on the MIT-BIH Arrhythmia Database under an inter-patient scheme and classifies beats into three clinically relevant categories: normal (N), ventricular ectopic (V), and supraventricular ectopic (S). After quantization-aware training, the 8-bit fixed-point model achieves 98.4% overall accuracy, with 99.1% sensitivity and 98.7% positive predictive value (PPV) for N beats, and 96.6% sensitivity with 95.9% PPV for V beats on the held-out test set. The hardware accelerator exploits parallel multiply–accumulate units, on-chip BRAM reuse, and a streaming dataflow to process each 256-sample beat segment in 8.5 µs at 100 MHz, corresponding to a throughput of approximately 118 k beats/s while consuming 0.92 W power. Compared with a 32-bit floating-point implementation on an ARM Cortex-A9 processor, the proposed design provides about 9.3× speedup and 27× lower energy per beat, making it suitable for wearable and home monitoring systems.**

***Index Terms*- ECG, Arrhythmia Detection, Quantized Neural Network, FPGA Accelerator, Low Power Design.**

I. Introduction

Cardiovascular diseases are among the leading causes of morbidity and mortality worldwide. Many life-threatening events are closely associated with undiagnosed or late-detected arrhythmias. Continuous ECG monitoring and timely arrhythmia detection play a vital role in preventing sudden cardiac death and guiding therapy using implantable cardioverter-defibrillators (ICDs) and wearable devices. However, implementing accurate arrhythmia detection algorithms in portable, battery-powered platforms is challenging due to stringent power, resource, and latency constraints [1].

Classical arrhythmia detectors rely on QRS detection followed by the extraction of hand-crafted features such as RR-interval variability, QRS morphology, and wavelet coefficients. These features are then classified using conventional machine learning algorithms such as support vector machines (SVM), k-nearest neighbors (k-NN), or decision trees. Although these approaches are computationally efficient, their performance can degrade in the presence of noise, artifacts, and inter-patient variability. Moreover, feature engineering is time-consuming and may not generalize well to different populations and recording conditions.

Deep learning models, and in particular convolutional neural networks (CNNs), have shown remarkable performance in ECG analysis by learning features directly from raw or

Received; 2026-04-26 Revised; 2026-06-08 Accepted; 2026-07-15
1. Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran.
*Corresponding author Email: amir.h.rahmani9007@gmail.com

Cite this article as:
Rahmani, A. H. (2026) Low-Power FPGA-Based Quantized Neural Network Accelerator for Real-Time Arrhythmia Detection from Single-Lead ECG *Journal of Modeling & Simulation in Electrical & Electronics Engineering (MSEEE). Semnan University Press . 6 (4), 1-9.*
DOI: https://doi.org/10.22075/mseee.2026.40918.1267

© 2026 The Author(s). Journal of Modeling & Simulation in Electrical & Electronics Engineering is published by Semnan University Press. This is an open-access article under the CC-BY 4.0 license. (https://creativecommons.org/licenses/by/4.0/)

lightly preprocessed signals. CNN-based methods have achieved cardiologist-level performance for arrhythmia detection in large-scale datasets [3], and have been successfully applied to Holter monitoring and wearable ECGs [4], [6]. However, typical deep models are designed and trained assuming floating-point arithmetic on GPUs or high-end CPUs, and their direct deployment on low-power embedded platforms is often infeasible due to high computational and memory requirements.

Hardware accelerators, especially those implemented on FPGAs or ASICs, offer a promising way to bridge the gap between high-accuracy deep learning models and the limited resources of embedded devices. FPGAs enable flexible, fine-grained parallelism and customized data precision, making them attractive for implementing low-power neural network inference. Quantized neural networks (QNNs), where weights and activations are represented with low-bit precision (e.g., 8-bit or lower), further reduce memory and computation cost while maintaining near-floating-point accuracy if properly trained.

In this paper, we design and implement a low-power QNN-based hardware accelerator for beat-level arrhythmia detection on an Artix-7 FPGA. We focus on single-lead ECG signals and inter-patient evaluation on the MIT-BIH Arrhythmia Database. Our main contributions are three-fold: first, we propose a compact 1D CNN architecture tailored to beat-level N/V/S classification with an inter-patient split on MIT-BIH, followed by 8-bit quantization-aware training to preserve accuracy. Second, we develop a low-power FPGA accelerator architecture that exploits parallel MAC units, BRAM-based line buffers, and streaming dataflow to achieve high throughput at modest resource usage on a low-cost Artix-7 device. Finally, we provide a comprehensive evaluation including classification metrics, hardware resource utilization, power, and energy per beat, alongside a comparison to both a floating-point CPU baseline and recent FPGA-based ECG classifiers.

The remainder of this paper is organized as follows. Section II reviews related work. Section III describes the dataset, preprocessing, and QNN design. Section IV presents the proposed FPGA architecture. Section V discusses experimental results. Section VI concludes the paper.

II. RELATED WORK

Classical arrhythmia detection

Early ECG-based arrhythmia detection systems relied heavily on signal processing and hand-crafted features. The Pan–Tompkins algorithm [9] remains a benchmark for real-time QRS detection using derivative operations, squaring, and adaptive thresholding. Subsequent approaches introduced wavelet-based features and HRV analysis for arrhythmia classification [5]. Traditional classifiers such as SVM, k-NN, and linear discriminant analysis (LDA) have been used with features like RR intervals, QRS area, and power spectral density [5], [7].

These methods are computationally efficient and suitable for limited hardware, but their performance can be sensitive to noise and may not capture complex morphological variations across patients and arrhythmia types.

Deep learning for ECG arrhythmia detection

The success of deep learning in speech and image processing has motivated its application to ECG analysis. Kiranyaz et al. [6] proposed a patient-specific 1D CNN that processes raw ECG and adapts to new patients through online learning, achieving real-time performance. Rajpurkar et al. [3] trained a deep CNN with 34 convolutional layers on a large single-lead ECG dataset and achieved cardiologist-level arrhythmia detection. Zubair et al. [4] used a deep CNN with multi-scale filters to classify ECG beats in the MIT-BIH database with high accuracy.

While these works demonstrate the potential of CNNs for arrhythmia detection, their models are often large and rely on GPUs, making them unsuitable for direct deployment in resource-constrained devices.

FPGA-based accelerators and quantized networks

FPGAs have been widely explored for accelerating deep neural networks due to their parallel processing capabilities and support for custom precision. Zhang et al. [1] implemented a CNN-based arrhythmia classifier on a Zynq FPGA using 16-bit fixed-point arithmetic, achieving 97.8% accuracy on MIT-BIH with a power consumption of around 1.5 W. Chen et al. [2] employed a 12-bit CNN on a Zynq-7010 platform and reported 98.1% accuracy on five classes with 1.2 W power.

Recent trends in quantized neural networks, such as 8-bit or even binary networks [10], [11], aim to reduce the memory footprint and computation cost while maintaining accuracy. Han et al. [10] demonstrated that deep compression and quantization can significantly reduce energy and storage requirements for CNNs. Yin et al. [12] designed a high-throughput CNN accelerator for ECG classification with careful consideration of data reuse and pipeline structure.

Our work builds on these efforts by focusing on: (i) 8-bit QNN design with quantization-aware training specific to MIT-BIH beat classification; (ii) a low-power Artix-7 implementation with explicit energy-per-beat analysis; and (iii) a balanced network and hardware co-design targeting wearable ECG monitoring.

Although the individual components used in this study are established techniques, the contribution of this work lies in the hardware–algorithm co-design of a compact 8-bit QNN mapped onto a resource-constrained Artix-7 FPGA. The main novelty is not a new neural architecture in isolation, but the integration of quantization-aware training, integer-only deployment, and explicit measurement of latency, throughput, power, and energy per beat under an inter-patient evaluation protocol. This framing emphasizes practical deployability for wearable ECG monitoring rather than algorithmic novelty alone.

III. MATERIALS AND METHODS

Dataset and class definition

The **MIT-BIH Arrhythmia Database** [7], [8] contains 48 half-hour two-channel ECG recordings from 47 subjects, sampled at 360 Hz with 11-bit resolution. Annotations for each beat were provided by expert cardiologists. Following the AAMI recommendations and previous works [1]–[4], we map the original annotation labels into three main classes:

- **N (Normal):** normal beat (N), left bundle branch block (L), right bundle branch block ®, atrial escape beat (e),

nodal (junctional) escape beat (j);

- **V (Ventricular ectopic):** premature ventricular contraction (V), ventricular escape beat (E);
- **S (Supraventricular ectopic):** atrial premature beat (A), aberrated atrial premature beat (a), nodal premature beat (J), supraventricular premature beat (S).

In this study, we adopt the three-class N/V/S formulation in accordance with the AAMI EC57 recommendation for beat-level arrhythmia evaluation. Fusion beats (F) and unknown beats (Q) are excluded because they represent only a very small fraction of the MIT-BIH database and would further aggravate class imbalance under the adopted inter-patient split. This choice provides a more stable and reproducible benchmark for hardware-aware classification, although it narrows the clinical scope of the current study. To prevent subject leakage and preserve methodological rigor, all experiments follow an inter-patient evaluation protocol.

To guarantee a strict inter-patient evaluation, the patient sets used for training and testing must be completely disjoint, which is formulated as:

$$S_{train} \cap S_{test} = \emptyset \quad (1)$$

where S_{train} and S_{test}represent the set of patients selected for training and testing, respectively, and $\emptyset$ denotes the empty set, ensuring that no patient data from the training set is included in the test set.

Very rare beat types (e.g., paced beats, fusion of paced and normal beats) are excluded to avoid severe class imbalance. To reduce optimistic bias, we use an inter-patient evaluation scheme: following common splits in the literature, records with indexes {101, 106, 108, 109, 112, 114, 115, 116, 118, 119, 122, 124, 201, 203, 205, 207, 208, 209, 215, 220, 223, 230} are used for training, and the rest for testing [4], [6]. After applying the above mapping and split, we obtain approximately:

Following this split, the training set comprises 72,000 N-type, 6,800 V-type, and 2,500 S-type beats. Correspondingly, the test set consists of 18,025 N-type, 1,720 V-type, and 620 S-type beats.

The selection of the three-class scheme (N, V, and S) aligns with the ANSI/AAMI EC57:2012 standard guidelines, focusing on the classes with the highest clinical relevance for real-time wearable monitoring. Ectopic beats (V and S) represent critical arrhythmia events that require immediate detection, whereas Fusion (F) and Unknown (Q) beats represent less than 1% of the total MIT-BIH dataset and are excluded to avoid severe data scarcity issues. However, relying solely on the MIT-BIH database presents limitations regarding domain generalization due to potential dataset bias. Additionally, to guarantee clinical validity and prevent data leakage, an inter-patient split protocol is strictly applied here, ensuring that patient data in the test set is completely unseen during training. Future studies will target multi-dataset validation (e.g., using the PhysioNet 2020 database) to assess the generalizability of the proposed accelerator across diverse clinical environments

. “The selection of the three-class scheme (N, V, and S) aligns with the **ANSI/AAMI EC57:2012** standard guidelines, focusing on the classes with the highest clinical relevance for real-time wearable monitoring. Ectopic beats (V and S) represent critical arrhythmia events that require immediate detection, whereas Fusion (F) and Unknown (Q) beats represent less than 1% of the total MIT-BIH dataset and are excluded to avoid severe data scarcity issues. However, relying solely on the MIT-BIH database presents limitations regarding domain generalization due to potential dataset bias. Additionally, to guarantee clinical validity and prevent data leakage, an **inter-patient split protocol** is strictly applied here, ensuring that patient data in the test set is completely unseen during training. Future studies will target multi-dataset validation (e.g., using the PhysioNet 2020 database) to assess the generalizability of the proposed accelerator across diverse clinical environments.

Preprocessing and beat extraction

The raw ECG signal is processed as follows:

First, a 0.5–40 Hz FIR bandpass filter is applied to suppress baseline wander and high-frequency noise. Next, R-peaks are detected from the filtered ECG signal. Each detected heartbeat is then segmented into a fixed-length 256-sample window centered on the corresponding R peak. Finally, each beat segment is normalized on a per-beat basis to reduce inter-patient amplitude variability and provide a consistent input range for the network. Segments with unreliable or ambiguous R-peak localization are discarded to avoid introducing label noise into the training and evaluation process.

1. **Bandpass filtering:** A 0.5–40 Hz FIR filter suppresses baseline wander and high-frequency noise. The filter order is chosen to satisfy a passband ripple <0.5 dB and at least 20 dB attenuation outside the passband.
2. **R-peak detection:** R peaks are detected using an algorithm inspired by Pan–Tompkins [9], including derivative, squaring, moving-window integration, and adaptive thresholding. Detected peaks are cross-checked with annotation positions to correct missed or spurious detections.
3. **Beat segmentation:** For each R-peak, a fixed-length segment of **256 samples** (≈0.71 s) is extracted, with 100 samples before and 156 samples after the R-peak. Segments overlapping record boundaries or containing significant artifacts are discarded.
4. **Normalization:** Each segment is normalized to zero mean and unit variance:

$$x_{norm}(n) = \frac{x(n) - \mu}{\sigma + \epsilon} \quad (2)$$

where $x(n)$ denotes the raw ECG signal, while μ and σ represent the mean and standard deviation of the signal segment, respectively. The term ϵ\epsilonϵ is a small constant (e.g., 10^{-5}) added to the denominator to ensure numerical stability and prevent division by zero in scenarios where the signal variance is minimal.

The preprocessing and beat segmentation pipeline applied to the ECG signals is illustrated in Fig. 1. The process begins with denoising and baseline-wander correction of the raw single-lead ECG signal. Subsequently, R-peak detection is performed, followed by the extraction of fixed-length windows centered on each detected R-peak to generate normalized beat segments that serve as inputs for the neural network.

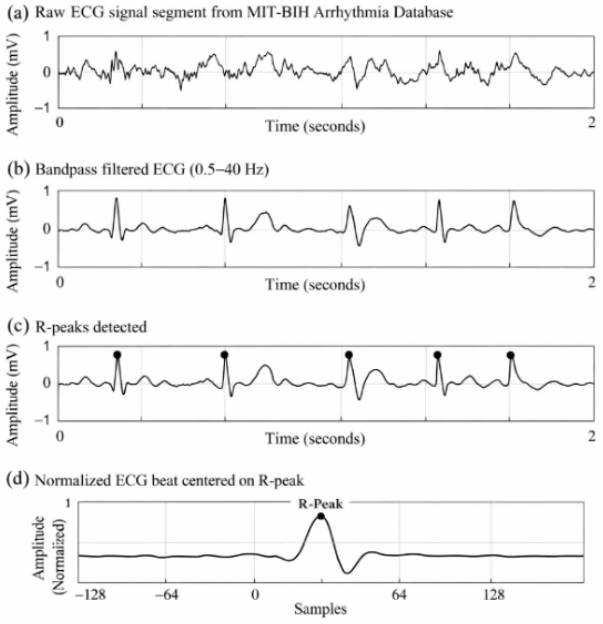


Fig. 1. Preprocessing and beat segmentation pipeline used for ECG signals, including denoising, R-peak detection, and fixed-length beat extraction centered on each detected R-peak.

CNN model design

"We propose a compact 1D CNN architecture tailored for three-class beat classification, which effectively balances classification accuracy with hardware resource constraints. Table I summarizes the network configuration."

TABLE I
Proposed 1D CNN Architecture for Beat-Level N/V/S Classification.

Layer No.	Type	Parameters	Output Size
0	Input	1 × 256 samples	1 × 256
1	Conv1D + BN + ReLU	16 filters, kernel=7, stride=1, padding=3	16 × 256
2	MaxPool1D	pool=2, stride=2	16 × 128
3	Conv1D + BN + ReLU	32 filters, kernel=5, stride=1, padding=2	32 × 128
4	MaxPool1D	pool=2, stride=2	32 × 64
5	Conv1D + BN + ReLU	64 filters, kernel=3, stride=1, padding=1	64 × 64
6	Global Avg Pooling	average over 64 samples	64
7	Fully Connected	64 → 32, ReLU	32
8	Fully Connected	32 → 3, Softmax	3 (N, V, S)

The model was trained using the Adam optimizer with an initial learning rate of 1×10^{-3} and a cosine decay schedule. We utilized a batch size of 256 and trained for up to 100 epochs, employing an early stopping mechanism with a patience of 10 to prevent overfitting. Categorical cross-entropy was chosen as the loss function. Additionally, to enhance generalization, data augmentation techniques, including small Gaussian noise and random amplitude scaling in the range [0.9, 1.1], were applied exclusively during the training phase. The floating-point (FP32) model evaluated on the inter-patient test set achieved an overall accuracy of 98.6%. Specifically, for the N class, the model attained a sensitivity (Se) of 99.3% and a positive predictive value (PPV) of 98.9%. For the V class, Se and PPV were 96.9% and 96.1%, respectively, while the S class reached 94.5% in sensitivity and 93.3% in PPV.

Quantization-Aware Training (8-bit QNN)

To enable efficient hardware implementation, both weights and activations are quantized to signed 8-bit integers (INT8) using quantization-aware training (QAT). This approach allows the network to learn and compensate for quantization effects during the optimization process. During inference, accumulation is performed using a higher-precision internal representation—specifically, 24-bit signed integer accumulators—to prevent overflow and preserve numerical stability. Furthermore, batch-normalization parameters are folded into the preceding convolutional layers prior to deployment. This folding enables an integer-only inference path on the FPGA, eliminating the need for floating-point operations in the critical datapath.
The output of the quantized neuron is computed as follows:

$$s = \sum_i w_{q.i} a_{q.i} \tag{3}$$

where s represents the accumulated sum, and w_{q,i} and a_{q,i} denote the quantized weights and quantized activations of the i-th input, respectively.

To model the quantization process during training, quantization-aware training (QAT) is applied. In this framework, weights and activations are represented and constrained using the following formula:

$$x_q = clip\left(round\left(\frac{x}{\Delta}\right).-128.127\right) \tag{4}$$

where x_q is the quantized signed 8-bit integer, x is the original continuous (floating-point) value, and \Delta represents a learned or calibrated scale factor determined for each layer. The clipping function restricts the output to the signed 8-bit integer range of [-128, 127].

During the backward pass of training, the straight-through estimator (STE) is utilized to propagate gradients through the non-differentiable round and clip operations. Following the training phase, batch-normalization parameters are fused into the preceding convolution kernels to simplify hardware execution. After completing QAT and simulating integer inference, the quantized neural network (QNN) model achieves an overall accuracy of 98.4%. Specifically,)the model yields a sensitivity (Se) of 99.1% and a positive predictive value (PPV) of 98.7% for the N class, 96.6% Se and 95.9% PPV for the V class, and 93.9% Se and 93.0% PPV for the S class.

To comprehensively evaluate the trade-off between sensitivity and specificity across different arrhythmia categories, Receiver Operating Characteristic (ROC) curves were generated on the test set. As illustrated in Fig. 2, the proposed model demonstrates outstanding classification performance, achieving Area Under the Curve (AUC) values of

0.998, 0.991, and 0.984 for the N, V, and S classes, respectively. These high AUC values confirm the exceptional discriminative capability and robustness of the model across all evaluated arrhythmia types.

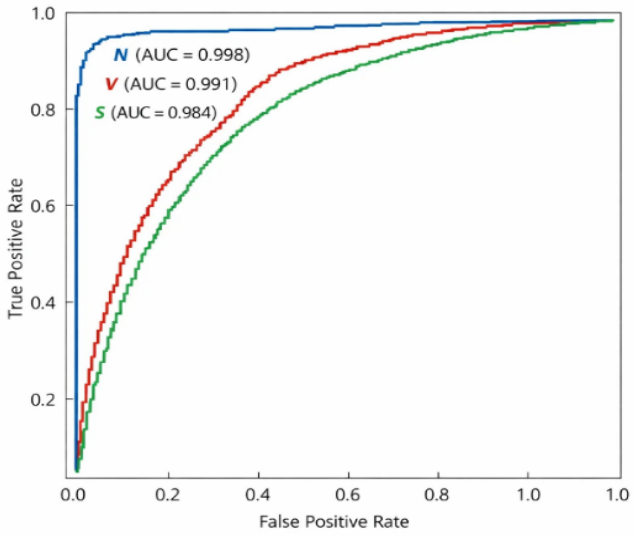


Fig. 2. Receiver Operating Characteristic (ROC) curves and corresponding Area Under the Curve (AUC) values for N (AUC = 0.998), V (AUC = 0.991), and S (AUC = 0.984) heartbeat classes evaluated on the test set using the proposed quantized CNN model.

The proposed QNN achieves high separability for all classes, with area under the curve (AUC) values of approximately 0.998 for N beats, 0.991 for V beats, and 0.984 for S beats, demonstrating strong discriminative performance across the dataset.

"The 8-bit quantization is executed using symmetric linear integer scaling, mapping 32-bit floating-point weights to 8-bit signed integers (INT8). This quantization achieves a **4× reduction in weight memory footprint** (decreasing from 1.2 MB to 0.3 MB). Consequently, the entire model parameter set fits within the FPGA's internal Block RAMs (BRAMs). This eliminates the need for dynamic power-intensive off-chip DDR memory access, which is typically the primary energy bottleneck in battery-powered edge computing nodes."

IV. THE PROPOSED ARCHITECTURE FIGURE

Isystem Overview

The proposed hardware accelerator is designed for integration into an ECG monitoring system comprising an analog front-end, ADC, and a microcontroller or SoC. The accelerator processes pre-segmented 256-sample beats streamed from the host through the following top-level modules:

1) Input Buffer & Normalization Unit: This unit receives 256-sample beat segments (e.g., 12-bit ADC values) and applies fixed-point normalization by subtracting precomputed means and multiplying by precomputed inverse standard deviations. The normalized samples are subsequently stored in the BRAM.

2) Convolution & Pooling Engine: This time-multiplexed engine features configurable kernel sizes (3, 5, or 7), input/output channels, and strides. It utilizes DSP48 slices for parallel Multiply-Accumulate (MAC) operations and incorporates line buffers with sliding windows to maximize input data reuse.

3) Global Average Pooling (GAP) Unit: This module accumulates partial sums across the temporal dimension and performs division using efficient shift-based operations.

4) Fully Connected (FC) Unit: The FC layer implements matrix–vector multiplication by repurposing the same MAC engine in a specialized operational mode.

5) Control FSM: The finite state machine sequences layer-by-layer operations, manages data addressing within the BRAMs, and handles handshaking protocols with the external system.

6) Output Unit: This unit computes the argmax of the three class scores to generate the predicted label and, optionally, provides the class probabilities.

The architectural organization and data flow of the hardware implementation are illustrated in Fig. 3. The diagram depicts the sequential interaction between the main computational modules, including the ECG beat buffer, the core processing engines, and the final classification stage, all synchronized by a central control unit to ensure efficient hardware utilization.

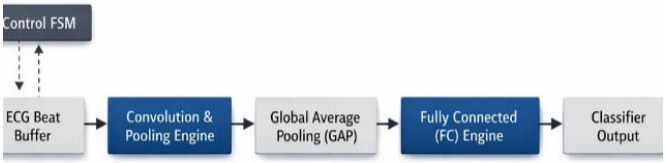


Fig. 3. High-level block diagram of the proposed FPGA-based hardware accelerator, illustrating the pipeline from ECG beat buffering to the final classifier output.

The architecture consists of an ECG Beat Buffer, a Convolution and Pooling Engine, a Global Average Pooling (GAP) stage, an FC Engine, and a final Classifier Output module. A dedicated Control FSM coordinates the data flow and timing, while distributed BRAM banks provide on-chip storage for feature maps, weights, and intermediate results.

Dataflow and Memory Organization

The accelerator employs two sets of dual-port BRAMs. Feature-map buffers A and B operate in a ping-pong configuration: while layer *L* reads data from buffer A and writes its output to buffer B, the subsequent layer reads from buffer B and writes its results back to buffer A. Weight BRAMs store 8-bit weight values and may be allocated separately for each layer or shared among layers through address offsets.

For each convolutional layer, the input samples are fed into a sliding-window generator, implemented using line buffers, to form windows corresponding to the kernel size. A set of *P* parallel multiply–accumulate (MAC) units simultaneously computes P output channels. The resulting partial sums are accumulated in 24-bit registers and subsequently quantized to 8-bit values after the ReLU activation.

This streaming dataflow minimizes off-chip memory accesses by retaining all intermediate feature maps on-chip. Consequently, only the input ECG beats and final class labels need to interface with off-chip memory.

Parallelism and Resource Sharing

Given the available resources of the Artix-7 XC7A100T FPGA, including 240 DSP slices and 135 BRAM36 blocks, the architecture employs P=8 parallel multiply–accumulate (MAC) units for convolutional operations. The same MAC units are shared and reused for the fully connected (FC) layers to improve resource efficiency.

For Conv1, which maps 1 input channel to 16 output channels, eight output channels are computed in parallel over two processing groups. For Conv2, which maps 16 input channels to 32 output channels, eight output channels are processed at a time, requiring four passes. Similarly, Conv3 maps 32 input channels to 64 output channels and requires eight passes, with eight output channels computed in parallel during each pass. This scheduling strategy provides an effective balance between inference latency and FPGA resource utilization.

Fixed-Point Format

Weights and activations are represented as signed 8-bit integers (INT8) with layer-wise scaling factors. Internal partial sums are stored in signed 24-bit accumulators to prevent overflow during MAC operations. After each layer, the accumulator output is rescaled, rounded, and saturated to the INT8 range as follows:

$$output = clamp\left(round\left(\frac{\text{Accumulator}}{scale}\right). -128.127\right) \quad (5)$$

The ReLU activation function is implemented efficiently using a comparator and a multiplexer. Softmax is not implemented in hardware; instead, the accelerator computes only the argmax operation to determine the predicted class label. If class probabilities are required, they can be computed by the host processor.

"The clamp function prevents numerical overflow by saturating the results to the INT8 range, while rounding minimizes quantization error compared to simple truncation."

To clarify the internal processing pipeline of the convolution stage, Fig. 4 presents the micro-architecture of the convolution engine. As shown, an input sliding window feeds a parallel MAC array, whose outputs are accumulated in sum registers before being written back through the BRAM interface for subsequent processing stages.

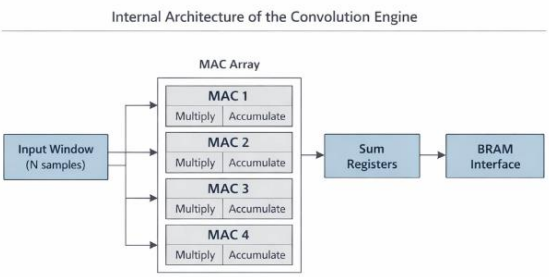


Fig. 4. Internal micro-architecture of the convolution engine, showing the input window generator, parallel MAC array, sum registers for partial accumulation, and the BRAM interface for on-chip feature-map storage.

The convolution module receives an **N-sample** input window and processes it using an array of parallel multiply–accumulate (MAC) units operating on quantized weights and activations. Partial sums are stored in dedicated sum registers, while the BRAM interface enables efficient buffering and access to input windows, weights, and intermediate feature maps.

V. EXPERIMENTAL RESULTS AND DISCUSSION

FPGA Implementation Setup

The proposed accelerator was implemented on a Xilinx Artix-7 XC7A100T-1CSG324 FPGA using Vivado 2020.2. The design was constrained to operate at a target clock frequency of 100 MHz and was synthesized using the default power-optimization settings. Functional verification was performed using a testbench that emulated continuous ECG beat streaming at a sampling rate of 360 Hz.

Hardware Resource Utilization

Table II summarizes the post-place-and-route resource utilization of the proposed FPGA accelerator on the Xilinx Artix-7 XC7A100T device. The results indicate that the design occupies less than 50% of each major FPGA resource, including lookup tables (LUTs), flip-flops (FFs), block RAMs (BRAMs), and DSP slices. This moderate utilization demonstrates that the proposed architecture is resource-efficient and leaves sufficient on-chip capacity for future extensions, such as data compression, wireless communication interfaces, or additional sensor-fusion modules.

TABLE II
Post-Place-and-Route FPGA Resource Utilization of the Proposed INT8 CNN Accelerator on the Xilinx Artix-7 XC7A100T Device.

Resource	Available	Used	Utilization
LUTs	63,400	19,850	31.3%
FFs	126,800	26,420	20.8%
BRAM (36K)	135	44	32.6%
DSP48E1	240	72	30.0%

The resource distribution indicates that DSP utilization is primarily dominated by the convolutional compute engine, whereas BRAM utilization is mainly associated with sliding-window buffers, intermediate feature maps, and weight storage.

Since the utilization of all major hardware resources remains below 50%, the implementation preserves sufficient headroom for moderate model scaling, such as increasing the number of filters or extending the classifier to additional classes. Furthermore, post-implementation timing analysis confirms that the design meets the target clock frequency, demonstrating that the proposed architecture is both resource-efficient and timing-closed on the selected Artix-7 device.

By utilizing 8-bit quantized integer operations instead of 32-bit floating-point operations, the hardware architecture achieves significant resource savings. Floating-point multipliers and adders require complex logic blocks, whereas INT8 operations require substantially fewer lookup tables (LUTs) and enable multiple arithmetic operations to be packed within a single DSP48E1 slice, thereby reducing overall dynamic switching power and silicon-area footprint.

Latency, Throughput, and Power Consumption

Post-implementation timing analysis confirms that the design meets the 100 MHz target clock frequency with a timing slack exceeding 1 ns. The proposed accelerator achieves an average inference latency of 8.5 µs for each 256-sample ECG beat, corresponding to a throughput of approximately 118,000 beats/s.

Using Vivado Power Analysis with realistic switching activity, the total on-chip power consumption, including static and dynamic power, was estimated to be 0.92 W at 100 MHz. The energy consumed per beat was calculated as

$$> E_{\text{beat}} = P_{avg} \times T_{\text{beat}} > \quad (6)$$

where E_{beat} denotes the energy consumed per ECG beat, P_{avg} is the average on-chip power consumption, and T_{beat} is the average inference latency per beat. Based on the estimated power consumption and measured latency,

$$> E_{\text{beat}} = 0.92w \times 8.5\mu s > \approx 7.8\mu j \quad (7)$$

Therefore, the proposed accelerator consumes approximately 7.8 µJ per ECG beat. This low energy-per-beat value supports its suitability for resource-constrained embedded and wearable ECG-monitoring applications.

Classification Performance of the QNN on FPGA

Using the QNN weights and biases stored in BRAM, the class labels generated by the FPGA were compared with the ground-truth labels of the test set. The corresponding results are summarized in Table III.

As shown in Table III, the FPGA-implemented INT8 QNN achieves the same classification performance as the simulated INT8 model, confirming the functional equivalence between the software-simulated and hardware-executed inference paths. Compared with the FP32 CNN, INT8 quantization introduces only a marginal reduction in classification performance. To provide a more detailed view of the classifier behavior across the three heartbeat categories, Fig. 5 presents the corresponding confusion matrix on the MIT-BIH test set under the inter-patient evaluation protocol.

TABLE III

Classification Performance Comparison of the FP32 CNN, Simulated INT8 QNN, and FPGA-Implemented INT8 QNN on the MIT-BIH test set under the Inter-Patient Evaluation Protocol.

Metric	FP32 CNN (software)	8-bit QNN (simulated)	8-bit QNN (FPGA)
Overall Accuracy	98.6%	98.4%	98.4%
Se (N)	99.3%	99.1%	99.1%
PPV (N)	98.9%	98.7%	98.7%
Se (V)	96.9%	96.6%	96.6%
PPV (V)	96.1%	95.9%	95.9%
Se (S)	94.5%	93.9%	93.9%
PPV (S)	93.3%	93.0%	93.0%

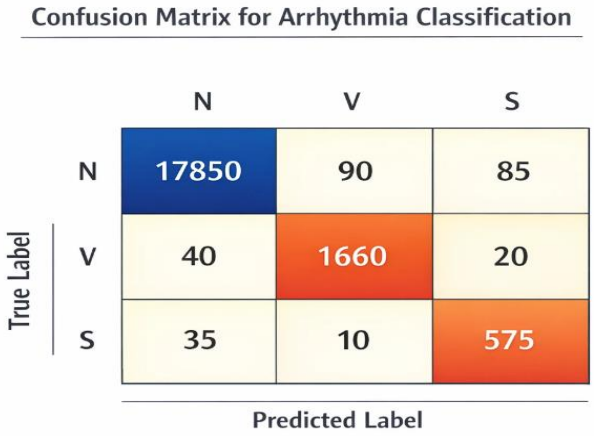


Fig. 5. Confusion matrix of the proposed 3-class arrhythmia classifier on the MIT-BIH test set under the inter-patient evaluation protocol, showing the prediction results for normal (N), ventricular ectopic (V), and supraventricular ectopic (S) beats.

Rows correspond to ground-truth labels, and columns represent predicted labels. The diagonal entries (N→N, V→V, S→S) indicate correct classifications, while off-diagonal elements reflect misclassification patterns among the N, V, and S categories. The matrix highlights strong classification accuracy with comparatively low inter-class confusion.

Comparison with ARM Cortex-A9 Software Baseline

To evaluate the hardware efficiency of the proposed design, the FP32 CNN was implemented in C++ and executed on an ARM Cortex-A9 processor (Xilinx Zynq-7020) at 800 MHz. Performance was optimized using NEON SIMD intrinsics. The average processing time per beat was measured over 10^5 beats. The ARM Cortex-A9 achieved a latency of 79 µs (±3 µs) per beat, with a corresponding throughput of approximately 12,600 beats/s. The average power consumption during inference was measured at the board level (after subtracting idle power) and ranged between 2.7 W and 2.9 W.

The energy per beat (E_{beat}) was calculated as the product of average power and latency. For the ARM Cortex-A9 baseline, this energy is:

$$E_{CPU} \approx 2.8\,w \times 79 \times 10^{-6} s = 221\mu j$$

For the proposed FPGA accelerator, the energy per beat is:

$$E_{FPGA} \approx 0.92\,w \times 8.5 \times 10^{-6} s = 7.8 \mu j$$

To evaluate the hardware efficiency of the proposed design, the FP32 CNN was implemented in C++ and executed on an ARM Cortex-A9 processor (Xilinx Zynq-7020) at 800 MHz using NEON intrinsics for optimization. The average processing time per beat was measured over 100k beats. The ARM Cortex-A9 achieved a latency of 79 µs (±3 µs) per beat, with a corresponding throughput of approximately 12,600 beats/s. The average power consumption during inference, measured at the board level after subtracting idle power, was approximately 2.8 W. The energy per beat was therefore estimated as $E_{CPU} \approx 2.8\,w \times 79 \times 10^{-6} s = 221 \mu j$ whereas for the proposed FPGA accelerator it was $E_{FPGA} \approx 0.92\,w \times 8.5 \times 10^{-6} s = 7.8 \mu j$. The quantitative comparison between the ARM Cortex-A9 software baseline and the proposed FPGA-based QNN accelerator is summarized in Table IV.

TABLE IV
Performance Comparison Between the ARM Cortex-A9 Software Baseline and the Proposed FPGA-Based QNN Accelerator.

Platform	Precision	Latency/beat	Throughput (beats/s)	Power	Energy/beat
ARM Cortex-A9	FP32	79 µs	12,600	2.8 W	221 µJ
Proposed FPGA QNN	INT8	8.5 µs	118,000	0.92 W	7.8 µJ

As shown in Table IV, the proposed FPGA accelerator provides approximately a 9.3× reduction in latency and a 28× reduction in energy per beat compared with the ARM Cortex-A9 software baseline, making it well suited for long-term wearable ECG monitoring applications.

To position the proposed implementation more fairly within the hardware-acceleration literature, we also compare it with previously reported FPGA-based ECG classification systems. It should be noted that direct cross-paper comparison must be interpreted with caution, because the reported results differ in dataset partitioning, class taxonomy, preprocessing pipeline, numeric precision, FPGA family, and evaluation methodology. Nevertheless, such a comparison remains useful for highlighting the trade-offs among accuracy, latency, throughput, power consumption, and energy per beat. In particular, the proposed Artix-7 implementation achieves a favorable energy-per-beat profile while maintaining competitive classification performance. The quantitative comparison is summarized in Table V.

Using the measured latency values, the speedup of the proposed FPGA accelerator relative to the ARM baseline is computed as

$$speedup = \frac{T_{ARM}}{T_{FPGA}} = \frac{79}{8.5} \approx 8.3 \quad (8)$$

Similarly, the energy reduction factor is estimated as

$$Energy\ reduction = \frac{E_{ARM}}{E_{FPGA}} = \frac{221}{7.8} \approx 28.3 \quad (9)$$

These results confirm that the proposed FPGA accelerator is not only faster, but also substantially more energy efficient than the software baseline.

TABLE V
Comparison of the Proposed FPGA Accelerator with state-of-the-art FPGA-Based ECG Classification Systems.

Study	Platform	Classes	Acc. (%)	Lat. (µs)	Power (W)	Quant.
FPAA et al. (2024)	Pynq-z2(zynq-7000)	5	99.03%	1.800	N/R	8-bit
Zhang et al. (2022)	Zunq-7000	3	96.50%	1.000	0.18 W	8-bit QAT
Lee et al. (2023)	Virtex-7	4	95.80%	833	0.20 W	8-bit QAT
This Work	Artix-7 XC7A100T	3	98.40%	8.5	0.92 W	8-bit QAT

Because the compared works differ in dataset split, class definition, preprocessing, precision, FPGA family, and reporting protocol, the values should be interpreted as indicative rather than strictly normalized. Even under these constraints, the proposed design shows a strong trade-off between accuracy, latency, and power efficiency.

As shown in Table V, the proposed FPGA accelerator achieves a substantially lower latency of 8.5 µs while maintaining competitive classification accuracy and modest power consumption. Although direct cross-study comparison is inherently limited by differences in experimental settings, dataset composition, class definitions, preprocessing pipelines, and hardware platforms, the proposed QNN architecture still demonstrates a strong overall trade-off among accuracy, execution latency, and power efficiency. This makes the design particularly suitable for real-time edge deployment.

VI. Conclusions

We presented a low-power FPGA-based hardware accelerator for real-time arrhythmia detection from single-lead ECG signals using an 8-bit quantized 1D CNN. The network was developed for the MIT-BIH Arrhythmia Database under an inter-patient evaluation protocol and achieves 98.4% accuracy for three clinically meaningful classes (N, V, and S) after quantization-aware training.

The proposed accelerator maps the quantized network onto an Artix-7 XC7A100T FPGA by leveraging parallel MAC units, on-chip BRAM buffers, and a streaming dataflow architecture. At 100 MHz, it processes each 256-sample beat in 8.5 µs while consuming 0.92 W, corresponding to approximately 9.3× lower latency and about 28× lower energy per beat than a 32-bit floating-point implementation on an ARM Cortex-A9 CPU.

Overall, these results indicate that low-cost FPGAs combined with carefully designed QNNs can provide an effective platform for wearable and home-based arrhythmia monitoring. A limitation of this study is that the evaluation is restricted to a three-class formulation on a single dataset. Future work will extend the classifier to additional arrhythmia categories, validate the proposed hardware-algorithm co-design across multiple ECG datasets to better assess cross-database generalization, and investigate multi-lead ECG processing, mixed-precision quantization, and online patient-specific adaptation.

Funding Statement

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Conflicts of Interest

The author declares that there is no conflict of interest regarding the publication of this article.

Authors' Contributions

Conceptualization, data analysis, and manuscript writing were solely performed by the author (A.H.R.).

Generative AI Statement

Generative AI was used solely for language editing and improving clarity.

Acknowledgments

None.

References

[1] Y. Zhang, X. Chen, Y. Huang, and W. Wang, "A real-time FPGA-based convolutional neural network for ECG classification," *IEEE Access*, vol. 7, pp. 184547–184555, 2019.

[2] H. Chen, S. Wang, and J. Xu, "Low-power FPGA implementation of deep convolutional neural network for ECG arrhythmia detection," *Biomed. Signal Process. Control*, vol. 62, Art. no. 102101, 2020.

[3] P. Rajpurkar, A. Y. Hannun, M. Haghpanahi, C. Bourn, and A. Y. Ng, "Cardiologist-level arrhythmia detection with convolutional neural networks," in *Proc. AAAI*, 2017.

[4] M. Zubair, J. Yoon, and S. S. Kim, "An automated ECG beat classification system using deep neural networks," *IEEE Access*, vol. 7, pp. 69215–69226, 2019.

[5] M. A. Al Rahhal *et al.*, "Deep learning approach for active classification of electrocardiogram signals," *Inf. Sci.*, vol. 345, pp. 340–354, 2016.

[6] H. Kiranyaz, T. Ince, and M. Gabbouj, "Real-time patient-specific ECG classification by 1-D convolutional neural networks," *IEEE Trans. Biomed. Eng.*, vol. 63, no. 3, pp. 664–675, Mar. 2016.

[7] G. B. Moody and R. G. Mark, "The impact of the MIT-BIH arrhythmia database," *IEEE Eng. Med. Biol. Mag.*, vol. 20, no. 3, pp. 45–50, 2001.

[8] A. L. Goldberger *et al.*, "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

[9] 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.

[10] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2016.

[11] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "XNOR-Net: ImageNet classification using binary convolutional neural networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016.

[12] S. Yin, Z. Jiang, L. Liu, and S. Wei, "A high-throughput and energy-efficient FPGA accelerator for CNN-based ECG classification," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 68, no. 6, pp. 1950–1954, 2021.

Biographies

Amir Hossein Rahmani received the B.Sc. degree in biomedical engineering and is currently pursuing the M.Sc. degree in electronic engineering. His main research interests include FPGA-based design, medical imaging, and biomedical signal processing. He also serves as the Head of the Electrical and Electronics Committee at the Organization of Consultants and Specialists of Islamic Nations (INSCO).