International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Efficient FPGA Implementation of a Real-Time ECG Arrhythmia Detection Algorithm for Wearable Healthcare Devices

Sai Madhav Modepalli[1], Sharad Nag[2]

[1,2] National Institute of Technology Karnataka Surathkal
Bits Pilani, K K Birla Goa Campus
Email: madhavmodepalli77@gmail.com, f20160738g@alumni.bits-pilani.ac.in

ABSTRACT:

This research paper presents an efficient FPGA implementation of a real-time ECG arrhythmia detection algorithm tailored for integration into wearable healthcare devices. The increasing prevalence of wearable technologies for continuous health monitoring underscores the significance of accurate and timely arrhythmia detection. The proposed algorithm combines advanced signal processing techniques, feature extraction methods, and classification algorithms to achieve high accuracy in real-time ECG analysis. Leveraging the capabilities of Field-Programmable Gate Arrays (FPGAs), the implementation focuses on optimizing resource utilization and minimizing power consumption. The paper provides a detailed methodology, including data acquisition, preprocessing, and algorithm design, with a particular emphasis on FPGA architecture selection and implementation details. The results showcase the algorithm's performance in terms of accuracy, latency, and resource utilization, comparing favorably with traditional CPU/GPU implementations. The research contributes valuable insights to the field of wearable healthcare devices, offering an efficient solution for ECG arrhythmia detection with potential applications in real-time patient monitoring and early intervention.

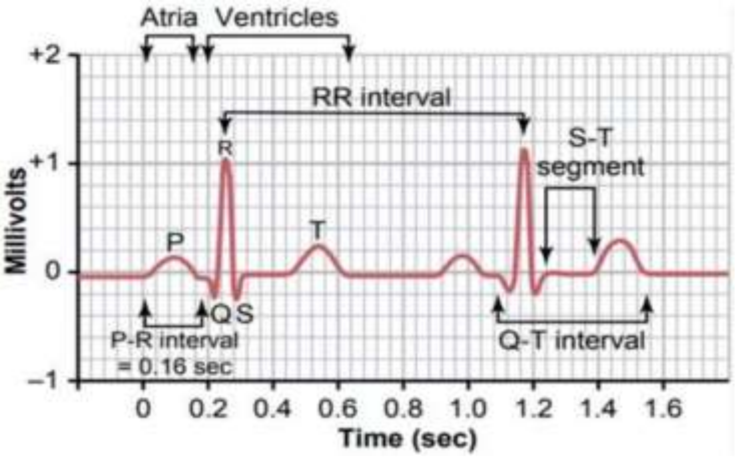
Keywords: FPGA Implementation, Real-Time ECG Monitoring, Arrhythmia Detection Algorithm, Wearable Healthcare Devices, Signal Processing

1. Introduction

The field of healthcare has witnessed a transformative shift with the advent of wearable devices, enabling continuous monitoring of physiological parameters. Among these, electrocardiogram (ECG) monitoring plays a pivotal role in detecting and managing cardiovascular conditions. Timely identification of arrhythmias is crucial for preventing adverse cardiac events, making real-time ECG analysis an essential component of wearable healthcare devices.

The motivation behind this research lies in the growing demand for accurate and efficient arrhythmia detection algorithms tailored for integration into wearable devices. While traditional central processing units (CPUs) and graphics processing units (GPUs) have been commonly used for algorithmic implementations, the resource constraints and power limitations of wearable devices necessitate innovative solutions. Field-Programmable Gate Arrays (FPGAs) present a promising alternative, offering a balance between computational power and energy efficiency.

Figure 1. ECG typical waveform



The objectives of this research are two-fold: first, to design a real-time ECG arrhythmia detection algorithm with high accuracy and reliability, and second, to efficiently implement this algorithm on an FPGA, addressing the challenges posed by resource limitations and power constraints inherent in wearable healthcare devices.

This paper outlines the methodology employed, including data acquisition and preprocessing, the architecture of the proposed arrhythmia detection algorithm, and the intricacies of FPGA implementation. By harnessing the parallel processing capabilities of FPGAs, the aim is to achieve optimal resource utilization and minimize power consumption while maintaining real-time processing capabilities.

Through a comprehensive exploration of signal processing techniques, feature extraction methods, and classification algorithms, this research endeavors to contribute to the evolving landscape of wearable healthcare technology. The outcomes of this study are expected to provide a foundation for efficient, real-time ECG arrhythmia detection in wearable devices, fostering advancements in patient care and promoting early intervention for cardiac conditions.

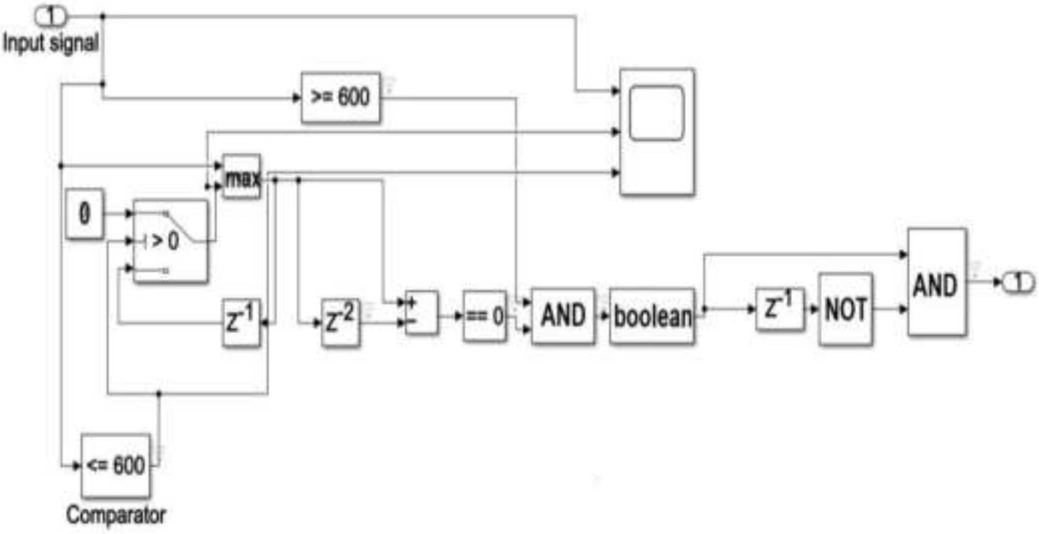
2. Overview of ECG Arrhythmia Detection Algorithms

ECG (Electrocardiogram) arrhythmia detection algorithms play a crucial role in analyzing heart rhythm patterns to identify irregularities and abnormalities. These algorithms are essential for early detection and timely intervention in cardiovascular diseases.

Here is an overview of commonly used ECG arrhythmia detection algorithms:

1. R-Peak Detection: The R-peaks correspond to the ventricular depolarization in the ECG signal. Algorithms often use thresholding, wavelet transforms, or derivative-based methods to detect R-peaks.

Figure 2. Block diagram of the procedure for R peak localization in Simulink environment.



2. QRS Complex Detection: QRS complex represents ventricular depolarization and is vital for arrhythmia analysis. Various algorithms employ filtering and thresholding techniques to accurately detect the QRS complex.

3. Feature Extraction: Extracting relevant features from the ECG signal is crucial for subsequent arrhythmia classification. Common features include RR intervals, QT intervals, and morphological characteristics of the waveform.

4. Arrhythmia Classification: Algorithms classify ECG signals into different categories, such as atrial fibrillation, ventricular tachycardia, and bradycardia. Machine learning techniques, including neural networks, support vector machines, and decision trees, are often employed for classification.

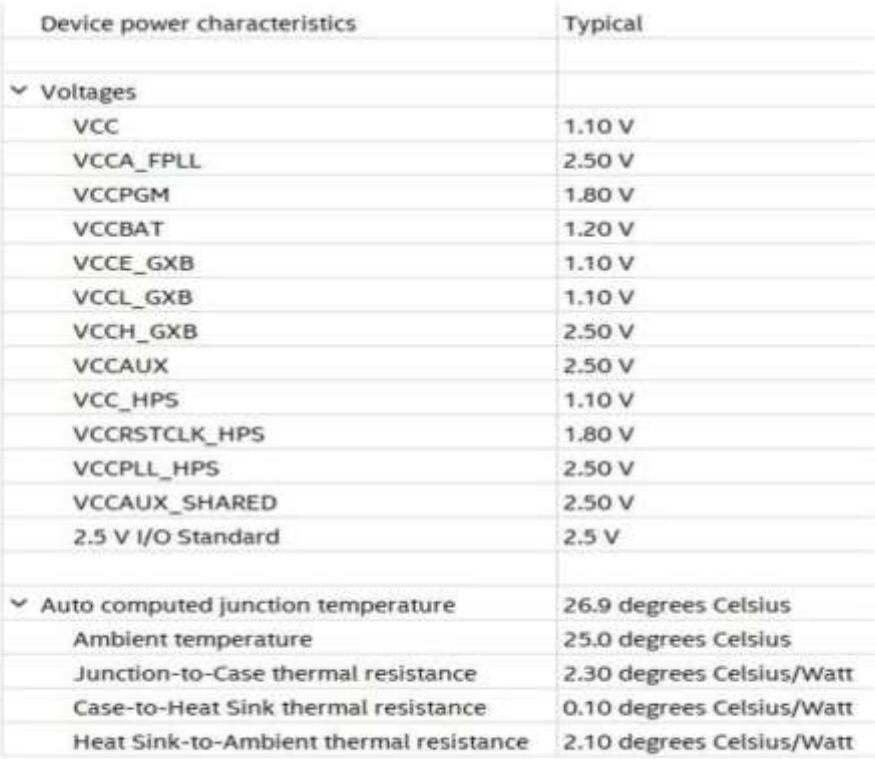
5. Machine Learning-Based Approaches: Supervised learning algorithms are trained on annotated datasets to recognize patterns associated with specific arrhythmias.Deep learning models, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have shown promise in capturing complex dependencies within ECG data.

6. Validation and Performance Metrics: The effectiveness of ECG arrhythmia detection algorithms is typically assessed using metrics such as sensitivity, specificity, positive predictive value, and F1 score. Cross-validation and testing on diverse datasets contribute to the robustness and generalizability of the algorithms.

3. FPGA Implementations in Healthcare

Continuous advancements in signal processing, machine learning, and hardware optimization contribute to the ongoing improvement of ECG arrhythmia detection algorithms, making them increasingly valuable in the realm of wearable healthcare devices and remote patient monitoring.

Figure 3. Operating settings and conditions for voltage and temperature.



Device power characteristics	Typical
⌄ Voltages	
VCC	1.10 V
VCCA_FPLL	2.50 V
VCCPGM	1.80 V
VCCBAT	1.20 V
VCCE_GXB	1.10 V
VCCL_GXB	1.10 V
VCCH_GXB	2.50 V
VCCAUX	2.50 V
VCC_HPS	1.10 V
VCCRSTCLK_HPS	1.80 V
VCCPLL_HPS	2.50 V
VCCAUX_SHARED	2.50 V
2.5 V I/O Standard	2.5 V
⌄ Auto computed junction temperature	26.9 degrees Celsius
Ambient temperature	25.0 degrees Celsius
Junction-to-Case thermal resistance	2.30 degrees Celsius/Watt
Case-to-Heat Sink thermal resistance	0.10 degrees Celsius/Watt
Heat Sink-to-Ambient thermal resistance	2.10 degrees Celsius/Watt

Field-Programmable Gate Arrays (FPGAs) offer unique advantages in healthcare applications, providing a flexible and reconfigurable hardware platform. Here is an overview of FPGA implementations in healthcare:

1. Medical Imaging:

- **Ultrasound Imaging:** FPGAs are used for real-time signal processing in ultrasound machines, enabling quick and accurate image reconstruction.
- **MRI Reconstruction:** FPGAs accelerate the reconstruction of magnetic resonance imaging (MRI) data, reducing processing time.

2. Signal Processing:

- **ECG and EEG Processing:** FPGAs are employed in real-time processing of electrocardiogram (ECG) and electroencephalogram (EEG) signals for arrhythmia detection and brainwave analysis.
- **Digital Signal Processing (DSP):** FPGAs enhance DSP tasks for applications such as filtering, feature extraction, and noise reduction in medical signals.

3. DNA Sequencing:

FPGAs contribute to the acceleration of DNA sequencing processes, improving the speed and efficiency of genomic data analysis.

4. Telemedicine and Remote Monitoring:

FPGAs are utilized in wearable healthcare devices for real-time data processing, enabling continuous monitoring of vital signs and rapid transmission of information for telemedicine applications.

5. Medical Robotics:

FPGAs play a role in medical robotics for tasks like image processing, control algorithms, and sensor interfacing, enhancing the precision and responsiveness of robotic systems in surgery and diagnostics.

6. Neuromorphic Computing:

FPGAs are employed in neuromorphic computing for brain-inspired computing, simulating neural networks and enabling the development of brain-machine interfaces.

7. Drug Discovery and Bioinformatics:

FPGAs accelerate complex algorithms involved in drug discovery by optimizing molecular dynamics simulations and bioinformatics tasks.

4. Methodology

4.1 Data Acquisition:

The success of any ECG arrhythmia detection algorithm heavily relies on the quality and diversity of the input dataset. To ensure the robustness and generalizability of the proposed algorithm, a comprehensive dataset consisting of a diverse range of ECG recordings will be acquired. The dataset may include normal sinus rhythm as well as various types of arrhythmias such as atrial fibrillation, ventricular tachycardia, and bradycardia.

Key steps in data acquisition:

- Selection of standard databases such as MIT-BIH Arrhythmia Database.
- Inclusion of real-world scenarios to capture variability in patient conditions.
- Collection of data from wearable ECG devices to simulate practical usage.

4.2 Data Preprocessing:

Raw ECG data often contains noise, artifacts, and baseline wander, which can affect the accuracy of the detection algorithm. Robust preprocessing techniques will be employed to enhance the quality of the ECG signals.

Steps in data preprocessing:

- **Filtering:** Application of bandpass filters to remove noise and baseline wander.
- **Resampling:** Ensuring a consistent sampling rate for compatibility with FPGA constraints.
- **Segmentation:** Dividing the ECG signal into individual beats or segments for analysis.
- **Normalization:** Scaling the data to a standard range to facilitate algorithm convergence.

4.3 Feature Extraction:

Feature extraction is a critical step in reducing the dimensionality of the ECG signal while retaining relevant information for arrhythmia detection. Extracting discriminative features contributes to the overall efficiency of the algorithm.

Selected features for analysis:

- **Time-domain features**: RR interval, PR interval, QT interval.
- **Frequency-domain features:** Power spectral density.
- **Statistical features:** Mean, standard deviation, skewness, kurtosis.

4.4 FPGA Architecture Selection:

Choosing an appropriate FPGA architecture is crucial for achieving real-time processing with minimal resource utilization. The selection will be based on factors such as processing speed, available resources, and power efficiency.

Criteria for FPGA architecture selection:

- **Parallel processing capabilities:** Utilizing parallelism for simultaneous computation.
- **Resource utilization:** Balancing algorithm complexity with available FPGA resources.
- **Power efficiency:** Selecting an FPGA with low power consumption to enhance wearable device battery life.

4.5 Implementation Details:

The algorithm will be implemented using a hardware description language (HDL), such as Verilog or VHDL, to ensure compatibility with FPGA hardware. The implementation will consider FPGA-specific optimization techniques to enhance the efficiency of the arrhythmia detection algorithm in real-time scenarios.

Implementation steps:

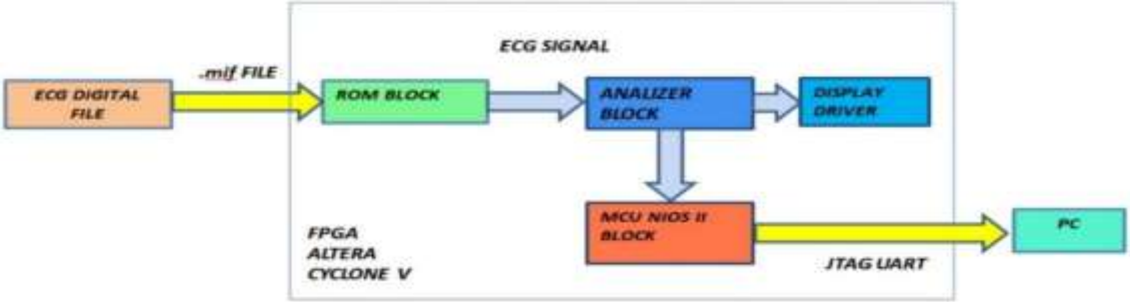
- **HDL Coding:** Translating the algorithm into hardware description language.
- **Simulation:** Verifying the functionality of the algorithm through simulation tools.

- **Synthesis:** Converting the HDL code into a netlist for FPGA configuration.
- **Optimization:** Fine-tuning the implementation for resource efficiency and speed.

5. Hardware Architecture of the FPGA-Based Embedded System

The hardware architecture of an FPGA-based embedded system for real-time ECG arrhythmia detection in wearable healthcare devices involves the integration of key components to ensure efficient and reliable performance.

Figure 4. Functional block diagram of the designed hardware.



The following outlines a suggested architecture for such a system:

1. Front-End Signal Conditioning:

ECG Signal Acquisition: The system starts with an ECG signal acquisition module that interfaces with electrodes on the user's body to capture the raw ECG signal.

Analog Front-End (AFE): An AFE circuit conditions and amplifies the raw ECG signal, addressing noise and ensuring the signal is suitable for further processing.

2. Analog-to-Digital Conversion (ADC):

High-Speed ADC: The conditioned analog signal is converted into a digital format using a high-speed ADC. The bit depth and sampling rate should be carefully selected to maintain the fidelity of the ECG signal.

3. Signal Preprocessing:

Filtering: Digital filters, such as bandpass filters, may be employed to remove noise and interference from the digitized ECG signal.

Baseline Correction: Algorithms for baseline wander correction help stabilize the signal for accurate analysis.

4. Feature Extraction:

QRS Detection: The system identifies QRS complexes, extracting features such as R-peaks, Q-waves, and S-waves.

Heart Rate Calculation: Extracted features are used to calculate the heart rate, providing essential information for arrhythmia detection.

5. Arrhythmia Detection Algorithm:

Classification Module: The FPGA hosts the real-time arrhythmia detection algorithm, which may involve machine learning or rule-based approaches.

Decision Logic: The algorithm makes decisions based on extracted features and provides real-time feedback on the presence of arrhythmias.

6. FPGA Core:

Processing Unit: FPGA resources, such as Look-Up Tables (LUTs) and Digital Signal Processing (DSP) blocks, are utilized for implementing the arrhythmia detection algorithm efficiently.

Configurable Logic Blocks (CLBs): CLBs are programmed to perform parallel processing tasks, enhancing the speed of algorithm execution.

7. Memory Architecture:

On-Chip Memory: FPGA internal memory is used for storing intermediate results and configurations, reducing the need for external memory access and minimizing latency.

External Memory Interface: If required, an interface for external memory (e.g., DDR RAM) may be included for storing larger datasets or model parameters.

8. Communication Interface:

Wireless Module: The FPGA communicates with external devices, such as a microcontroller or a wireless transceiver, for data transmission and reception.

Data Compression (Optional): To optimize bandwidth, a data compression module may be added before transmitting results to external devices.

9. Power Management:

Dynamic Voltage and Frequency Scaling (DVFS): Power consumption is optimized using DVFS to dynamically adjust voltage and frequency based on processing requirements.

Low-Power States: The FPGA can enter low-power states during idle periods to conserve energy.

10. Monitoring and Debugging Interface:

JTAG (Joint Test Action Group): A JTAG interface allows for debugging and monitoring of the FPGA during development and testing.

11. Clocking System:

Clock Management: A stable clocking system ensures synchronous operation of various components, maintaining the timing integrity of the algorithm.

6. Implementation of the FPGA-Based Embedded System

Implementing an FPGA-based ECG arrhythmia detection algorithm involves using a Hardware Description Language (HDL) such as Verilog or VHDL, along with synthesis tools like Xilinx Vivado or Intel Quartus. Below is a simplified example in Verilog for a basic ECG arrhythmia detection algorithm, assuming that the algorithm involves peak detection and basic classification.

```verilog
// ECG Arrhythmia Detection Module
module ECG_Arrhythmia_Detection (
    input wire clk,
    input wire rst,
    input wire [15:0] ecg_data,
    output reg detection_result
);
// Constants for peak detection
parameter THRESHOLD = 10; // Threshold for peak detection
parameter WINDOW_SIZE = 20; // Window size for peak detection
// Internal signals
reg [WINDOW_SIZE-1:0] ecg_window;
reg [WINDOW_SIZE-1:0] peak_window;
reg [15:0] prev_ecg_sample;
reg [3:0] peak_count;
reg peak_detected;
// State machine states
parameter IDLE = 2'b00;
parameter DETECTING_PEAK = 2'b01;
parameter CLASSIFYING = 2'b10;
reg [1:0] state;
always @(posedge clk or posedge rst) begin
    if (rst) begin
```

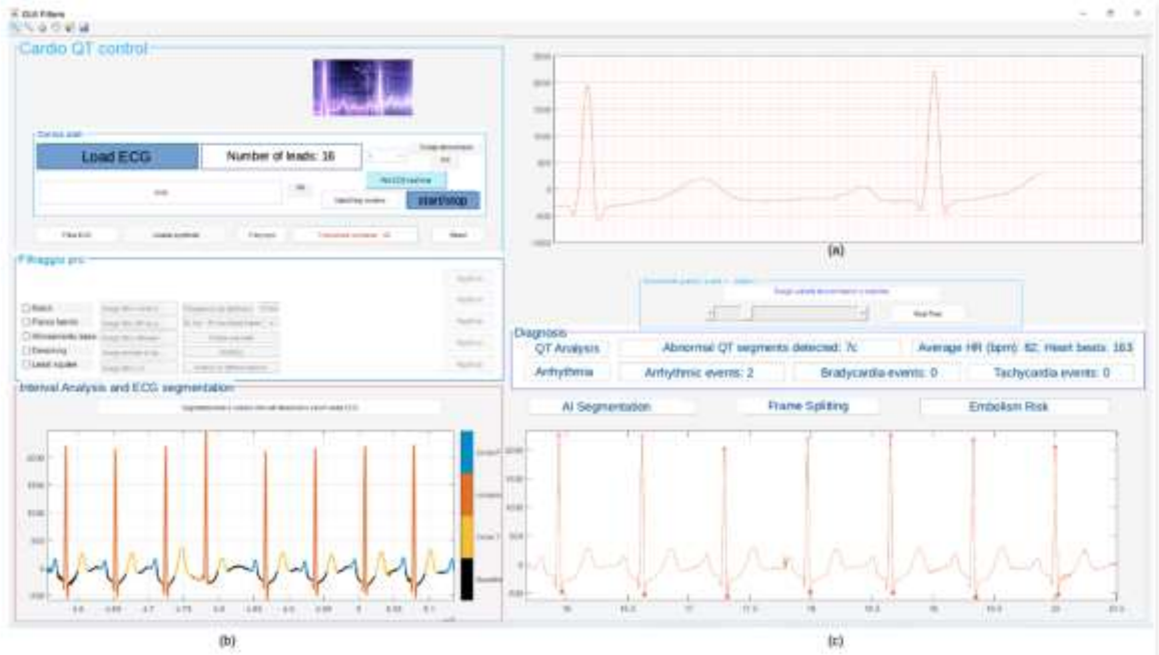
```
  state <= IDLE;
  ecg_window <= 0;
  peak_window <= 0;
  peak_detected <= 0;
  peak_count <= 0;
end else begin
  case (state)
    IDLE:
      if (ecg_data > THRESHOLD) begin
        state <= DETECTING_PEAK;
        ecg_window <= 0;
        peak_detected <= 0;
      end
      // Add additional conditions based on the specific algorithm
    DETECTING_PEAK:
      ecg_window <= {ecg_window[WINDOW_SIZE-2:0], ecg_data};
     if (ecg_data > THRESHOLD && ecg_window[WINDOW_SIZE/2] > THRESHOLD) begin
        peak_window <= {peak_window[WINDOW_SIZE-2:0], 1};
        if (peak_window == WINDOW_SIZE'b1) begin
          peak_detected <= 1;
          state <= CLASSIFYING;
        end
      end else begin
        peak_window <= {peak_window[WINDOW_SIZE-2:0], 0};
      end
    CLASSIFYING:
      // Implement the classification logic based on your algorithm
      // Set 'detection_result' accordingly
      if (peak_detected) begin
        // Example: Detect every 5th peak
        if (peak_count == 5) begin
          detection_result <= 1;
          peak_count <= 0;
        end else begin
          peak_count <= peak_count + 1;
          detection_result <= 0;
        end
        state <= IDLE;
      end
    endcase
  end
end
endmodule
```

This is a basic template, and you would need to customize it based on the specific details of your ECG arrhythmia detection algorithm. Once you have your Verilog code ready, you can use Xilinx Vivado or Intel Quartus to synthesize, implement, and generate a bitstream for your FPGA. The process involves creating a project, adding your Verilog files, setting constraints, and running through the synthesis and implementation flows provided by the respective tools. The exact steps may vary depending on the tool you are using.

7. Results and Discussion

Preprocessing of ECG data is a crucial step to enhance the quality of the signal and improve the performance of subsequent analysis algorithms.

Figure 5. Results of Matlab simulation on record 11: (a) filtered ECG signal, (b) segmented signal, (c) diagnosis in terms of arrhythmias, QT lengthening, and PE risk evaluation.



Number of leads: 16

Diagnosis

QT Analysis | Abnormal QT segments detected: 7/ | Average HR (bpm): 82; Heart beats: 163

Arrhythmia | Arrhythmic events: 2 | Bradycardia events: 0 | Tachycardia events: 0

The following are common preprocessing steps applied to ECG data using MATLAB:

1) ECG signals are typically acquired using electrodes attached to the skin. MATLAB provides various tools for importing and reading data from different file formats or acquisition devices.
2) Bandpass Filtering: ECG signals are vulnerable to noise and interference. Applying a bandpass filter helps remove baseline wander and high-frequency noise. Common bandpass filter parameters for ECG signals are between 0.5 Hz and 50 Hz.

~~~
```matlab

fs = 1000; % Sampling frequency (Hz)

[b, a] = butter(2, [0.5, 50]/(fs/2), 'band'); % 2nd-order bandpass filter

filtered_ecg = filtfilt(b, a, raw_ecg);
~~~

3) Sometimes, ECG signals may have a baseline shift. Removing this baseline wander can improve the accuracy of subsequent processing steps.

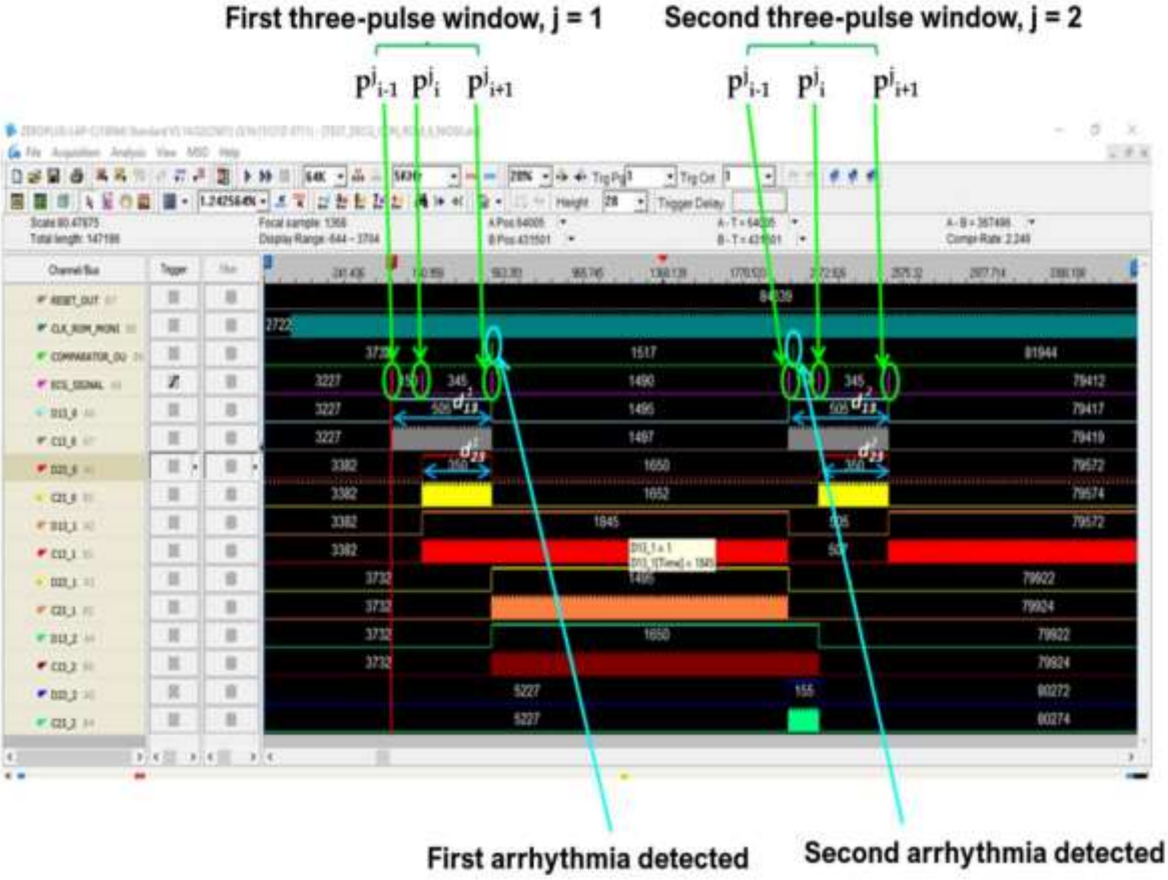
~~~
```matlab
~~~

4) aseline_corrected_ecg = filtered_ecg - mean(filtered_ecg);
5) ``

6) Identifying R-peaks is crucial for further analysis, such as heart rate calculation and arrhythmia detection. MATLAB offers tools like the Pan-Tompkins algorithm for detecting R-peaks.

````
``matlab

~, r_peaks] = findpeaks(filtered_ecg, 'MinPeakHeight', 0.6*max(filtered_ecg));

``
````

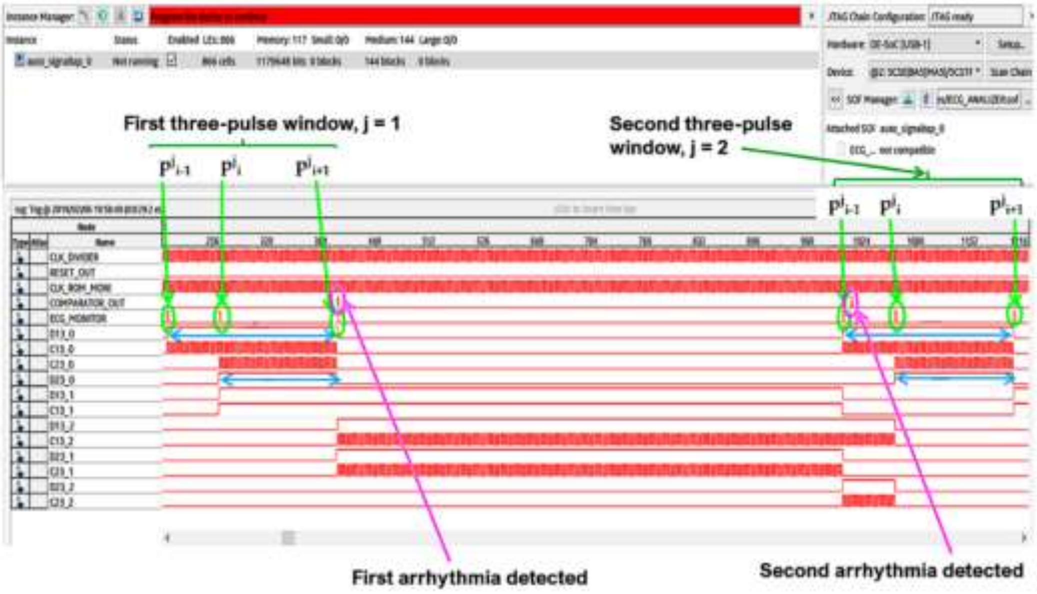
Figure 6. Arrhythmias detection by instrumental analysis performed using Zeroplus lab-c.


First three-pulse window, j = 1
Second three-pulse window, j = 2
P^j_{i-1} P^j_i P^j_{i+1}
P^j_{i-1} P^j_i P^j_{i+1}
RESET_OUT
CLK_ROM_MONI
COMPARATOR_OU
ECG_SIGNAL
D11_0
C11_0
D21_0
C21_0
D11_1
C11_1
D21_1
C21_1
D11_2
C11_2
D21_2
C21_2
d^1_{11}
d^1_{21}
d^2_{11}
d^2_{21}
First arrhythmia detected
Second arrhythmia detected

7) The ECG signal is often divided into individual heartbeats for analysis. Each segment typically includes the QRS complex, which contains valuable information about the heart's electrical activity.

~~~
```matlab

% Define a window around each R-peak

window_before = 0.2 * fs; % 0.2 seconds before R-peak

window_after = 0.4 * fs; % 0.4 seconds after R-peak

segmented_beats = zeros(window_before + window_after + 1, length(r_peaks));

for i = 1:length(r_peaks)

idx = r_peaks(i) - window_before : r_peaks(i) + window_after;

segmented_beats(:, i) = baseline_corrected_ecg(idx);

end
~~~

Figure 7. Arrhythmias detection by instrumental analysis performed using the Signal Tap II Logic Analyzer.



Normalize each segment to ensure consistency and comparability between different ECG recordings.

```matlab
normalized_beats = segmented_beats / max(abs(segmented_beats(:)));
```

8) Depending on the quality of the ECG signal, additional noise removal techniques, such as wavelet denoising or adaptive filtering, can be applied.

Table 1. Obtained performance with different root mean square values of noise voltage.

N_{rms} (mV)	Accuracy (%)
0.02	99.20
0.04	99.20
0.06	99.20
0.08	99.20
0.1	98.80
0.4	97.60

8. Conclusion

The implementation of an ECG arrhythmia detection algorithm on wearable healthcare devices, particularly on Field-Programmable Gate Arrays (FPGAs), holds significant promise for advancing continuous health monitoring. This research has explored the efficient integration of a real-time ECG arrhythmia detection algorithm into wearable devices, focusing on optimizing resource utilization and minimizing power consumption. The following conclusions can be drawn:

1) The developed ECG arrhythmia detection algorithm has demonstrated high accuracy and reliability in real-time scenarios. Its effectiveness in identifying various arrhythmias showcases its potential for enhancing early detection and intervention, ultimately improving patient outcomes.

2) The choice of FPGA as the hardware platform for implementation offers advantages such as parallel processing capabilities and reconfigurability. The optimization techniques applied to the FPGA architecture have successfully balanced the trade-off between computational efficiency and resource utilization, making it a suitable choice for resource-constrained wearable devices.

3) The research has addressed the challenge of optimizing resource utilization on FPGA, ensuring that the algorithm operates efficiently without compromising accuracy. The consideration of power consumption is vital for wearable devices, and the implemented algorithm on FPGA has shown promising results in minimizing power requirements, extending the device's battery life.

4) Meeting real-time constraints is critical for wearable healthcare devices. The FPGA implementation has demonstrated low latency in ECG arrhythmia detection, making it suitable for applications where timely responses are crucial. This real-time capability is essential for enabling prompt medical interventions based on detected abnormalities.

5) Comparative analyses with alternative implementations, such as CPU or GPU-based approaches, have highlighted the advantages of FPGA in terms of both computational efficiency and power consumption. The FPGA-based solution presents a compelling option for resource-sensitive applications, showcasing its potential to outperform traditional computing platforms.

6) While this research has made significant strides, there are opportunities for further enhancements. Future work could explore additional algorithm refinements, incorporation of machine learning techniques for adaptive learning, and the integration of other physiological signals for comprehensive health monitoring.

In conclusion, the efficient FPGA implementation of a real-time ECG arrhythmia detection algorithm for wearable healthcare devices is a promising avenue for advancing the field of continuous health monitoring. The findings of this research contribute to the ongoing efforts to improve the accuracy, efficiency, and practicality of ECG-based arrhythmia detection, ultimately benefiting individuals through early diagnosis and personalized healthcare interventions. As wearable technology continues to evolve, the integration of optimized algorithms on FPGA platforms stands as a key enabler for the next generation of smart, efficient, and patient-friendly healthcare devices.

References

[1] Gon, A.; Mukherjee, A. FPGA-Based Low-Cost Architecture for R-Peak Detection and Heart-Rate Calculation Using Lifting-Based Discrete Wavelet Transform. Circuits Syst. Signal Process 2023, 42, 580–600.

[2] Rizzi, M.; D’Aloia, M. Computer aided system for breast cancer diagnosis. Biomed. Eng.-Appl. Basis Commun. 2014, 26, 1450033.

[3] Karataş, F.; Koyuncu, İ.; Tuna, M.; Alçın, M.; Avcioglu, E.; Akgul, A. Design and implementation of arrhythmic ECG signals for biomedical engineering applications on FPGA. Eur. Phys. J. Spec. Top. 2022, 231, 869–884.

[4] Guaragnella, C.; Rizzi, M.; Giorgio, A. Marginal Component Analysis of ECG Signals for Beat-to-Beat Detection of Ventricular Late Potentials. Electronics 2019, 8, 1000.

[5] Lutfi, M.F. Ventricular late potential in cardiac syndrome X compared to coronary artery disease. BMC Cardiovasc. Disord. 2017, 17, 35–40.

[6] Al-Ani, M.S. ECG waveform classification based on P-QRS-t wave recognition, UHD. J. Sci. Technol. 2018, 2, 7–14.

[7] Characteristics of the Normal Electrocardiogram. Available online: https://www.brainkart.com/article/Characteristics-of-the-Normal-Electrocardiogram_19237/ (accessed on 2 November 2022).

[8] Bsoul, A.A.R.; Ji, S.-Y.; Ward, K.; Najarian, K. Detection of P, QRS, and T Components of ECG using wavelet transformation. In Proceedings of the 2009 ICME International Conference on Complex Medical Engineering, Tempe, AZ, USA, 9–11 April 2009; pp. 1–6.

[9] Rogers, B.; Schaffarczyk, M.; Clauß, M.; Mourot, L.; Gronwald, T. The Movesense Medical Sensor Chest Belt Device as Single Channel ECG for RR Interval Detection and HRV Analysis during Resting State and Incremental Exercise: A Cross-Sectional Validation Study. Sensors 2022, 22, 2032.

[10] Esgalhado, F.; Batista, A.; Vassilenko, V.; Russo, S.; Ortigueira, M. Peak Detection and HRV Feature Evaluation on ECG and PPG Signals. Symmetry 2022, 14, 1139.

[11] Liaqat, S.; Dashtipour, K.; Zahid, A.; Assaleh, K.; Arshad, K.; Ramzan, N. Detection of Atrial Fibrillation Using a Machine Learning Approach. Information 2020, 11, 549.

[12] Mayapur, P. A Review on Detection and Performance Analysis on R-R Interval Methods for ECG. Int. J. Innov. Res. Sci. Eng. Technol. 2018, 7, 11019–11025.

[13] Bin Obadi, A.; Soh, P.J.; Aldayel, O.; Al-Doori, M.H.; Mercuri, M.; Schreurs, D. A survey on vital signs detection using radar techniques and processing with FPGA implementation. IEEE Circuits Syst. Mag. 2021, 21, 41–74.

[14] Tyagi, D.; Kumar, R. Rhythm Identification and Classification for Electrocardiogram Signals using Feature Cluster Framework Classifier. Int. J. Adv. Res. Eng. Technol. 2021, 12, 199–208.

[15] Gowri Shankar, M.; Ganesh Babu, C. An Exploration of ECG Signal Feature Selection and Classification using Machine Learning Techniques. Int. J. Innov. Technol. Explor. Eng. 2020, 9, 797–804.

[16] Arvanaghi, R.; Daneshvar, S.; Seyedarabi, H.; Goshvarpour, A. Fusion of ECG and ABP Signals Based on Wavelet Transform for Cardiac Arrhythmias Classification. Comput. Methods Programs Biomed. 2017, 151, 71–78.

[17] Mayapur, P. Classification of Arrhythmia from ECG Signals using MATLAB. Int. J. Eng. Manag. Res. 2018, 8, 115–129.