

# VLSI-Based ECG Compression Framework for Wearable Sensor Applications

Harsha S<sup>1</sup>, Dr. Radha B L<sup>2</sup>

<sup>1</sup>MTech student, Dept of ECE, Bangalore Institute of Technology, India

<sup>2</sup>Professor, Dept of ECE, Bangalore Institute of Technology, India

doi.org/10.64643/IJIRTV12I4-184309-452

**Abstract**—In wireless body area networks (WBANs), continuous monitoring of biomedical signals such as ECG leads to the generation of large data volumes, which significantly increases transmission energy consumption. To address the challenges of storage and power efficiency, this work introduces a lossless ECG compression approach. The method combines Run-Length Encoding (RLE) with Golomb–Rice coding, forming a hybrid algorithm that improves compression performance. The proposed scheme is confirmed on the MIT-BIH arrhythmia database, where it achieves a compression ratio of 3.0. Furthermore, a dedicated VLSI architecture of the algorithm is realized in 90 nm CMOS technology, showing a power consumption of 153.27  $\mu$ W at an operating frequency of 120 MHz and a supply voltage of 1.2 V, while occupying only 28.47 mm<sup>2</sup> of chip area.

**Keywords**— Biomedical Signal Processing, Data Storage Reduction, Electrocardiogram (ECG), Golomb–Rice Coding, Lossless Compression, Low-Power VLSI, Run-Length Encoding (RLE), Wireless Body Area Networks (WBAN), 90 nm CMOS.

## I. INTRODUCTION

Electrocardiogram (ECG) signals are the most important biomedical signals used in clinical practice for monitoring the electric activity of the heart. Wearable sensor nodes implemented for continuous health monitoring generate significant volumes of data [1]. Since nearly 72% of their energy is consumed in wireless transmission [2], power efficiency becomes a critical design challenge to extend battery life. In present IoT-enabled WBAN systems, ECG signals are typically acquired and then digitized for transmission to a remote base station, where processing is performed off-node [3]. To reduce the transceiver activity and thereby lower power usage, on-node data compression is an effective strategy [3], [4].

Although lossy compression methods provide higher compression ratios (CR), they typically cause errors during signal reconstruction. Several lossy approaches reported in [1], [5]–[7] rely on complex feature extraction from specific signal domains. However, these methods demand heavy preprocessing to retain clinically significant features while minimizing distortion, which increases computational overhead [8]. Furthermore, domain transformations introduce additional storage requirements and latency [1].

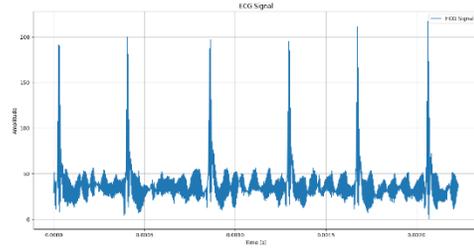


Fig. 1. Sample ECG signal acquired for analysis.

In contrast, lossless compression ensures exact signal recovery. Approaches such as prediction-based coding with Golomb–Rice encoding [9] and adaptive linear prediction with variable-length encoding [10] have been explored. VLSI realizations of adaptive linear predictors combined with Golomb–Rice entropy coding are discussed in [11]. Power-gating strategies especially in sensing have been proposed for low-power FPGA-based designs [12]. However, many of these methods remain computationally demanding, resulting in higher area overhead and increased consumption in power, which aren't ideal for wearable sensor nodes. In addition, the technique proposed here minimizes hardware complexity while retaining clinically relevant ECG information, thereby making it scalable for multi-channel monitoring.

It also provides a framework adaptable to different biomedical signals beyond ECG, offering flexibility for future WBAN applications. With these improvements, the proposed compression architecture strikes a balance between energy efficiency, storage reduction, and reconstruction accuracy, supporting long-term, real-time monitoring in wearable healthcare systems.

## II. PROPOSED SYSTEM

### A. Designed Scheme

In the proposed methodology, the first-order derivative  $D(n)$  is computed from two successive ECG samples, i.e., the difference between the present and the previous sample values. The resulting derivative values are typically centered around zero, as illustrated in Fig. 1, which depicts the histogram of  $D(n)$  for ten ECG signals of 1-minute duration each from the MIT-BIH database. However, because of the presence of characteristic ECG peaks such as the P-wave, R-peak, and T-wave, variations in amplitude occur, leading to differences in  $D(n)$  across different regions. The algorithm has been tested using digitized ECG inputs both with and without preprocessing. A flowchart summarizing the lossless compression approach is presented in Fig. 2.

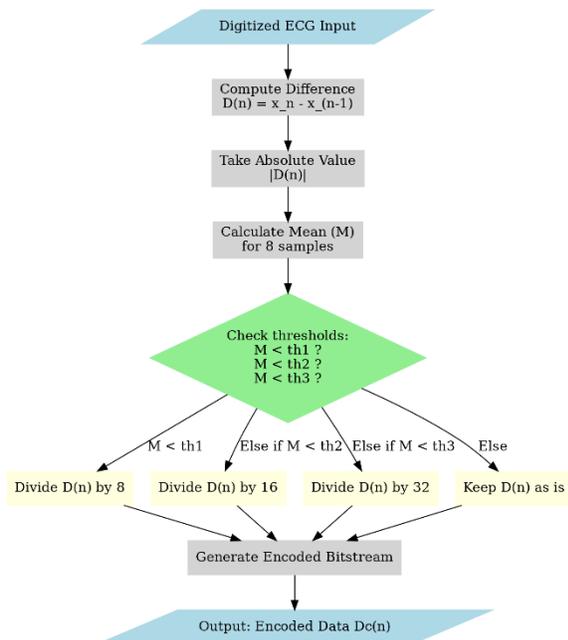


Fig. 2. Developed lossless compression scheme for ECG signals.

The process begins by evaluating the first derivative of ECG samples as:

$$D(n) = x_n - x_{n-1}$$

This operation compresses the amplitude range compared to the original digitized ECG signal. From the derived samples, a packet length of 8 is selected, which helps in capturing different ECG regions such as the high-amplitude QRS complex. The packet mean is calculated to incorporate both amplitude transitions and abrupt variations which may arise due to noise.

To further reduce the data range, a scaling operation is applied using divisors determined by the mean of the absolute derivative values  $|D(n)|$ . The mean  $M$  of the  $i$ th packet is given by:

$$M = \frac{1}{8} \times \sum_{j=i}^{i+7} |D(n)|_{j,i}$$

The obtained mean is now compared between three predefined threshold levels, which correspond to low-, medium-, and high-amplitude regions (with  $th3 > th2 > th1$ ). Depending on the region, different divisors are selected as outlined in Fig. 2. Finally, the scaled packet  $D1(n)$  is forwarded to the encoder, which generates the compressed bitstream  $Dc(n)$ .

### B. Bitstream Generation

Golomb–Rice coding is a widely adopted lossless compression method, particularly effective when the input data contains relatively small amplitude values [11]. In this scheme, a divisor which the power of two is used to generate the quotient and remainder. The quotient is encoded using unary representation, while the remainder is expressed in binary form. To differentiate between the two, a stop bit is inserted, with the bit length of the remainder depending directly on the chosen divisor. Since the ECG amplitude has already been reduced by computing  $D(n)$ , Golomb–Rice coding becomes an efficient choice for handling adaptive region-based divisors, especially when these divisors can be expressed as powers of two. From an implementation viewpoint, such divisors allow Golomb–Rice coding to be realized with minimal hardware complexity and cost.

The mathematical representation of the quotient and remainder for divisor parameters  $k=3,4,5$  is expressed as:

$$\text{Quotient} = \frac{D(n)}{2^k}, k = 3, 4, 5$$

$$\text{Remainder} = D(n) \bmod 2^k, k = 3, 4, 5$$

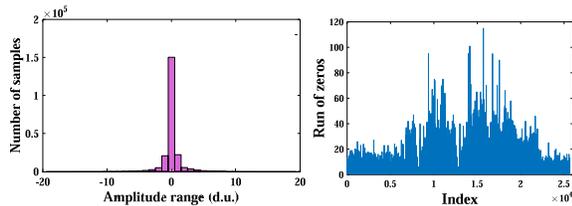


Fig. 3. (a) Histogram distribution for quotients in  $D1(n)$ .  
(b) Distribution of run of zeros for quotients in  $D1(n)$ .

The selection of the parameter  $k$  is carried out by evaluating the mean amplitude of  $|D(n)|$ , denoted as  $MR$ , across three distinct regions of the ECG signal: high-amplitude, medium-amplitude, and low-amplitude zones. The value of  $MR$  serves as a reference to determine the most suitable quotient values for  $D(n)$ . Accordingly, three  $k$ -values are chosen for these regions based on the relation:

$$k = \log_2(MR)$$

Fig. 3(a) illustrates the distribution of quotient values obtained from ten ECG signals, each of 1-minute duration, sourced from the MIT-BIH database. The analysis clearly indicates that a substantial portion of the quotients are zero, which directly points to the possibility of achieving a higher compression ratio (CR). To gain further insight, consecutive zero quotients were studied, as presented in Fig. 3(b). The results reveal that long runs of zeros, frequently extending beyond a length of 25, are observed in the dataset. Such behavior provides an excellent opportunity for optimization through run-length coding (RLC), which is more efficient in handling repeated symbols than using Golomb–Rice coding alone. Therefore, in the proposed hybrid scheme, RLC is applied to encode the successive zero quotients, while Golomb–Rice coding is dedicated to non-zero quotients. This dual-coding approach effectively balances compression performance with computational simplicity, making it highly suitable for real-time ECG signal processing in portable healthcare devices.

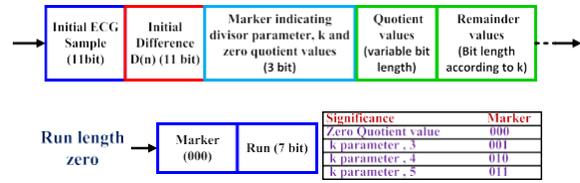


Fig. 4. Data packaging in proposed compression scheme.

### C. Packet Formation

The ECG signals collected in this project are taken from the MIT-BIH Arrhythmia Database available on the PhysioNet platform. Each ECG record contains digitized samples collected at a frequency of 360 Hz with 11-bit resolution. In our implementation, Python scripts were used to read the ECG samples directly from the database files (.dat and .hea). The signals were normalized and converted into integer arrays, which formed the raw input for the compression system. This preprocessing ensured compatibility with subsequent steps like subtraction, run-length detection, and coding.

The first step in compression is to reduce redundancy between consecutive samples. ECG signals are slowly varying except at sharp QRS complexes. Therefore, instead of directly encoding raw sample values, the difference of the current sample and the previous sample is computed. Mathematically, this can be expressed as:

$$d[n]=x[n]-x[n-1]$$

Fig. 4 illustrates the data packaging structure, highlighting the use of markers in the proposed compression method. In run-length encoding, when the quotient in  $D1(n)$  is zero, it is represented by the marker ‘000’, followed by the number of consecutive zeros. For nonzero cases, the remainder The values are concatenated with the quotient terms in the output stream, with their bit length determined by the run count and the  $k$ -parameter. The nonzero samples are further encoded using Golomb–Rice coding, where the quotient part is represented in unary form the expressed in the binary representation of the remainder. To maintain uniqueness, all markers begin with “0”, ensuring clear distinction from the unary quotient codes.



B. FPGA Implementation using Verilog and Vivado

After validating the algorithm in Python, the design have been translated into hardware using Verilog HDL. The system architecture included:

- Difference Generator – Subtracts current and previous samples.
- Accumulator & Zero Detector – Counts consecutive zeros.
- Run-Length Counter – Measures run lengths.
- Conditional Shifter & Golomb–Rice Encoder – Performs quotient and remainder computation.
- Output Bit Assembler – Forms compressed bitstream.

The Verilog modules were synthesized and implemented on Xilinx Vivado, targeting an FPGA board for verification. Simulation waveforms confirmed correct address generation, run-length detection, and encoded output generation. By running the design on FPGA, the real-time feasibility of the compression scheme was validated.

- preprocessing
- Proposed method (Difference + Run-Length + Golomb–Rice)

B. Hardware Simulation in Vivado & Cadence

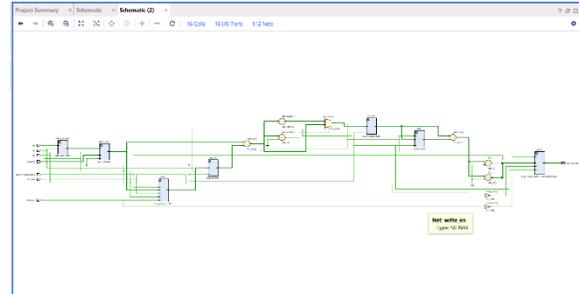


Fig. 6. Hardware schematic of the proposed design in Vivado.

The Verilog implementation of the algorithm is simulated using the Xilinx Vivado to verify functionality at the register-transfer level (RTL). The testbench fed binary ECG samples (converted from PhysioNet data) into the design. The schematic representations in 90 nm and 45 nm provide insights into the scalability of the architecture, highlighting the improvements achieved in terms of energy efficiency and performance at advanced technology nodes.

IV. PERFORMANCE ANALYSIS AND RESULTS

A. Compression Performance Analysis

The effectiveness of the developed compression The MIT-BIH (arrhythmia database) was used to verify the method, with ECG signals digitized at 360 Hz. To assess performance, the compression ratio (CR) is employed as the primary evaluation metric, which can be expressed as:

Here,  $Y_0$  represents the bitstream length of

$$CR = \frac{Y_0}{Y_1}$$

the original ECG input signal, while  $Y_1$  denotes the bitstream length of the corresponding compressed signal. The results of compression were compared across different methods:

- Direct Golomb–Rice coding without

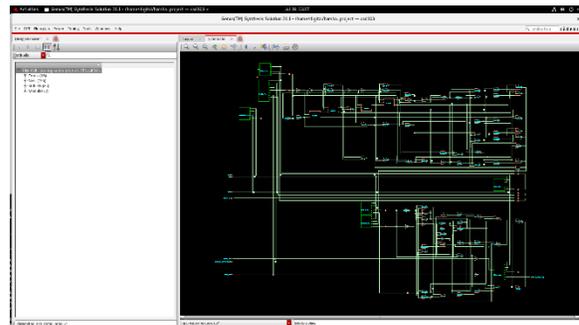


Fig. 7. Schematic designed in 90 nm technology using Cadence.

The schematics were implemented and simulated using Cadence tools, ensuring accurate transistor-level design representation. Both 90 nm and 45 nm CMOS technologies were explored to compare performance variations. The designs provide insight into the impact of technology scaling on power, delay, and area efficiency.

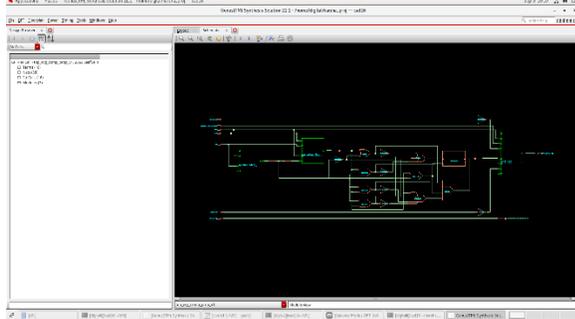
C. FPGA Implementation Results

The FPGA synthesis results from Vivado showed that

the proposed architecture required a small number of logic elements:

- Slice LUTs and Registers: The design used only a few hundred resources, well within the capacity of low-cost FPGAs.
- Timing: Maximum operating frequency was ~120 MHz.
- Power: Estimated dynamic power was in the order of a few milliwatts, and is therefore ideal for the portable medical applications.

Fig. 8. Schematic designed in 45 nm technology using



Cadence.

These results confirm that the architecture is compact and power-efficient, making it practical for wearable ECG devices.

#### D. Simulation Waveform Discussion

The Vivado waveforms clearly demonstrated the internal working of the modules:

- During initial clock cycles, the input ECG samples were processed by the difference generator, producing many zeros.
- After counting the zeros, the accumulator transferred the run-length to the Golomb–Rice encoder once the sequence ended
- The encoded quotient and remainder were assembled into a final bitstream.
- The output bitstream length was significantly shorter than the original input.

By expanding the waveform over small windows of 2–4 cycles, it was possible to trace the complete sequence from input sample to compressed output, ensuring correctness of the hardware implementation.

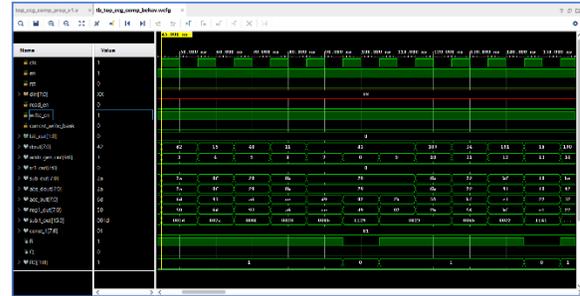


Fig. 9. FPGA simulation and implementation results showing the generation of Q and R

#### E. Hardware Implementation and Prototyping

For prototyping, the proposed architecture was first deployed on a Zync FPGA board based on the Zynq UltraScale+ MPSoCs platform, where the generation of quotient (Q) and remainder (R) corresponding to ECG samples from the MIT-BIH database was successfully demonstrated. The compression architecture was also realized in both 90 nm and 45 nm CMOS technologies using the FEL library. Functional verification of the design have been carried out using the Synopsys Vivado tool, while gate-level netlists were generated through Synopsys Design Compiler. Placement and routing were performed using ICC, with the critical path and clock tree highlighted in green and yellow, respectively. At an operating frequency of 120 MHz, the architecture exhibited power consumption as estimated by Synopsys tools. The FPGA synthesis confirmed stable operation up to 120 MHz, which is more than sufficient for real-time ECG applications since the sampling rate is only 360 Hz. Furthermore, power estimation using the Power Compiler at 25 °C and 1.2 V supply voltage reported a power consumption of 153.27  $\mu$ W, while the design occupied a silicon area of 28.47 mm<sup>2</sup> (NAND2 equivalent). Table 1 provides a comparative analysis of the proposed work against existing schemes. The results clearly indicate that the proposed design achieves superior compression ratio (CR) and demonstrates notable improvements in terms of power efficiency, area reduction.

Table 1. Comparative Performance Analysis

Reference	[1]		
	90nm	45nm	90nm
Process Technology	90nm	45nm	90nm
Supply Voltage	1.2 V	1.2 V	1.2 V
Power	18.78 $\mu$ W	0.783 $\mu$ W	153.27 $\mu$ W
Area	0.0051 mm <sup>2</sup>	0.0031 mm <sup>2</sup>	28.475 mm <sup>2</sup>
Compression Type	Lossless	Lossless	
Compression Ratio (CR)	2.91	3	
Operating Frequency	100 MHz	120 MHz	

## V. CONCLUSION

This project presents an efficient ECG compression system developed and validated on both software and hardware platforms. ECG signals from the PhysioNet database were preprocessed and compressed using differential coding, run-length encoding, and Golomb–Rice coding, achieving a compact binary representation. Python-based simulations showed that preprocessing improved the compression ratio to approximately 3.0 for 2048 samples, aligning with reported literature. The hardware design in Verilog was verified in Vivado, and then in Cadence with simulations confirming the correct operation of all modules, including the difference generator, accumulator, shifter, and encoder. FPGA synthesis indicated a lightweight, power-efficient design capable of real-time operation. Overall, the system effectively balances compression performance, hardware simplicity, and energy efficiency, making it suitable for ECG storage and transmission in constrained environments.

## REFERENCES

- [1] Jitumani Sarma and Rakesh Biswas, “A VLSI-Based Hybrid ECG Compression Scheme for Wearable Sensor Node,” *IEEE Sensors Letters*, vol. 6, no. 4, pp. 1–4, April 2024
- [2] M. H. Chowdhury and R. C. C. Cheung, “Reconfigurable Architecture for Multi-lead ECG Signal Compression with High-frequency Noise Reduction,” *Scientific Reports*, vol. 9, art. no. 17233, 2019.
- [3] A. Rajani and A. S. Sankar, “VLSI Implementation of Lossless ECG Compression Algorithm” *IRJET*, vol. 8, no. 10, pp. 1212–1214, Oct. 2021.
- [4] G. Erna, V. Pelluru, S. Shetty, and B. Raj, “FPGA Implementation of Hybrid ECG Compression and Decompression Method to Improve the Compression Ratio,” *Journal of Circuits, Systems and Computers*, May 2025.
- [5] W. Singh, “Biopotential Acquisition Unit for Energy-efficient Wearable Health Monitoring System with On-chip Data Compression,” *IET Circuits, Devices & Systems*, 2018.
- [6] C. Gungor, “A 1.2 nW Analog Electrocardiogram Processor Achieving...,” 2021.
- [7] K. Bannajak et al., “Signal Acquisition-Independent Lossless ECG Compression Using Prediction Error-based Adaptive Linear Prediction,” *PMC* (open access), 2023.
- [8] L. Zhang C. Tan, and H. Wu, “A Novel Blaschke Unwinding Adaptive Fourier Decomposition-based Signal Compression Algorithm with Application on ECG Signals,” *arXiv preprint*, Mar. 2018.
- [9] F. C. Bauer, D. R. Muir, and G. Indiveri, “Real-time Ultra-low Power ECG Anomaly Detection using an Event-driven Neuromorphic Processor,” *arXiv preprint*, Nov. 2019.
- [10] H. Mamaghanian, N. Khaled, and D. Atienza, “Development and evaluation of multi-lead wavelet-based ECG delineation algorithms for embedded wireless sensor nodes,” *IEEE Trans. Inf. Technol. Biomed.*
- [11] A. Dogan et al, “Multi-Core Architecture Design for Ultra-Low-Power Wearable Health Monitoring Systems,” *IEEE Trans. Integrated Circuits of CAD and Systems*
- [12] M. Elgendi, A. Mohamed, and R. Ward, “Efficient ECG Compression and QRS Detection for e-Health Applications,” *Scientific Reports*, 2017.
- [13] M. Elgendi, A. Al-Ali, A. Mohamed, and R. Ward, “Improving Remote Health Monitoring: A Low-Complexity ECG Compression Approach,” *Diagnostics*, 2018.
- [14] Y. Ziran, G. Guojun, H. Jiang, and M. Shuangwu, “Research and Improvement of ECG Compression Algorithm Based on EZW,” *Comput. Methods Programs Biomedicine*, 2017.