Chapter 5
ACLT-Based QRS Detection and ECG Compression Architecture

5.1 Introduction

Ultra-low power medical devices are imperative in the era of the IoT. Healthcare sensors capture vital physiological data for monitoring and diagnosing patients. Holter monitors is a case in point where it records and monitors continuous ECG data for 24 h. They are constrained by power consumption since they need to operate for an extended period continuously. On the other hand, IoT healthcare platform enables minimum local processing and transfers data to cloud-connected servers that help resolve drawbacks of Holter monitors and similar devices. Cloud platforms provide easy access for doctors to continuously follow up on their patients. Various platforms of IoT architectures for healthcare were proposed as in [60, 61]. IoT healthcare connects patients, doctors, and devices according to the philosophy shown in Fig. 5.1.

IoT infrastructure extends from sensors, communicating devices up to central servers which incorporate efficient devices [62]. IoT platform challenges result from system engineering that involves signal acquisition, local processing, transmission, central processing, and generating feedback [63]. Each of these stages has challenges, especially with increasing numbers of connected devices.

ECG is one of the most vital signals in IoT healthcare devices. ECG, which represents the electrical activity of the heart, is used as a prime tool to monitor and diagnose cardiac diseases due to the non-invasive nature of ECG sensors and the accuracy of mapping between ECG signals and heart physical activity. ECG is utilized in cardiac arrhythmia prediction and detection by extracting ECG intervals, amplitudes, and wave morphologies of the different components such as the P, QRS, and T waves [23]. The basis for extracting such parameters depends on the accurate real-time delineation of the ECG wave components. The development of real-time and accurate delineation methods is crucial for abnormal ECG signals that occur with different types of cardiac diseases.

T. Tekeste Habte et al., *Ultra Low Power ECG Processing System for IoT Devices*, Analog Circuits and Signal Processing,
https://doi.org/10.1007/978-3-319-97016-5_5

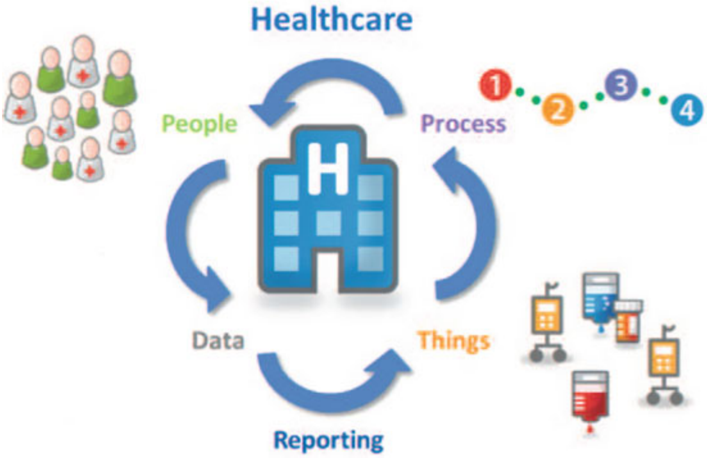


Fig. 5.1 IoT healthcare platform

The QRS complex, which is a principal component of the cardiac cycle, is used as a reference and represents the depolarization of ventricles in the heart. Its amplitude rises to 1 or 2 mV above or below the isoelectric line for normal beats and can go several times larger for abnormal beats. The time required for the ventricles to depolarize defines the QRS width or interval where it typically lasts between 80 and 120 ms [22]. QRS detection is a key for automatic delineation techniques. Various signal processing of QRS detection techniques have been proposed in the literature. Time domain thresholding along with filtering (first derivative, second derivative, both derivatives, matched filter, etc.) are some of the techniques that are suitable for real-time implementation [24–26]. In [24] Pan and Tompkins algorithm (PAT), which is one of the most widely researched and implemented techniques, was proposed, since it is robust in detecting QRS [12, 64]. Other methods that provide enhanced accuracy are based on the spectral analysis of the ECG signal. In [27–30], wavelet transform is presented as a tool to analyze ECG signals. As a part of the spectral analysis techniques, discrete Fourier transform has been reported in the literature to detect the QRS complex [31]. Empirical mode decomposition and Hilbert transform have been introduced to improve the analysis of the QRS detection of nonlinear and non-stationary ECG signals [32, 33].

Processed ECG data or extracted features in the IoT platform are transmitted wirelessly. Wireless data transmission is the most energy-hungry part in IoT devices. One of the effective ways in reducing energy consumed in wireless transmitters is to reduce the data transmitted through data compressors. In healthcare applications, lossless compression during transmission is a primary choice for reliability issues. Lossless ECG compressor architectures were reported in [65, 66]. Some recent data-compression schemes focused on lossy compression since it provides a high compression ratio [67]; however, it is less reliability when compared to lossless techniques. Lossy techniques have a high compression ratio in the range 2× up to 15×. However, lossless compressors provide a compression ratio range of 1× up to 3×.

Another option of reducing transmitted or processed data is decreasing the number of samples. In [68] a non-uniform time sampling technique is proposed with an adaptive sampling rate to reduce the energy consumption of the sampling process. Such a scheme is applicable to slowly varying signals. In [69] compressed sensing is presented as a potential technique for reducing the sample count, which is advantageous in reducing the overall power dissipation.

General-purpose micro-controllers could be the central processing unit of an IoT device. However, existing micro-controllers have an active power dissipation of greater than 100 μW and a leakage power of greater than 1 μW [70, 71], which is much higher power dissipation than that of custom ASIC solutions. Henceforth, the main reason to have a custom HW solution is to enable ultra-low power operation. The objective of this chapter is to present an ECG processing and compression architecture that will help IoT medical devices to achieve ultra-low power operation and to minimize the data needed to be transmitted to minimize power consumption. Operating at an ultra-lower power would enable the device to be powered by an energy harvester that generates power in the order of μW [72]. In this chapter, a multiplier-less ECG QRS detection architecture, which is based on a single transformation, is presented. Moreover, a compression technique based on first-derivative is proposed. The proposed QRS detection architecture consumed a 6.5 nW when implemented in 65 nm low-power process.

The remaining part of the chapter is organized as follows: Section 5.2 provides a summary of existing QRS detection techniques, Sect. 5.3 contains the full description of the proposed QRS detection architectures, Sect. 5.5 presents performance evaluation and results, Sect. 5.6 discusses the compressor comparison with literature, and Sect. 5.7 concludes the chapter.

5.2 Summary of QRS Detection and Compressor Architectures

5.2.1 Summary of QRS Detection Architectures

QRS detection is challenging due to the following reasons. ECG (being low amplitude in nature) is contaminated by noise and artifacts, such as electrode noise, motion artifacts, muscle noise, power-line interference, ADC quantization noise, and noise in acquisition devices. Moreover, QRS waves have wide morphological variations among different people with different health conditions. Several QRS detection architectures have been reported in literature each having its own merits and demerits. Here are some of the commonly existing architectures.

A. Discrete Wavelet Transform QRS detection based on quadratic spline wavelet transform is reported in [73]. Even though the system achieves high sensitivity and predictivity (99.31% and 99.7%) for QRS detection when validated using MIT-BIH database, its implementation is so complex that requires scale-3 wavelet transforms and maximum modulus recognition. Its operating power consumption is 0.85 μW.

B. Differentiation and Adaptive Thresholding In [64] a QRS detection architecture (QRS detection using differentiation, moving average, and squaring) is reported. Dynamically adaptive thresholds are applied to a squared ECG signal in order to detect QRS peaks. The system is optimized for an ultra-low power application that reduces computational complexity, however, still uses hardware-intensive operations such as multiplication and division.

5.2.2 Summary of ECG Compression Architectures

Several ECG compression architectures have been proposed and some of them are summarized below.

Fan Architecture Fan architecture for lossy ECG compression is reported in [74]. Fan is initially proposed in [75]. It operates by drawing the longest possible straight line between the starting sample and the ending sample, in such a way that during the reconstruction of samples, the error is less than the maximum specified error value, ϵ.

Lossless-Compressor Based on Linear Slope Predictor A low-power ECG compressing architecture, based on linear slope predictor, is reported in [65]. Moreover, it includes a fixed-length packaging-scheme for serial transmission. The architecture was implemented in 0.35 μm technology and achieves a compression ratio of 2.25×, at a power consumption of 535 nW, from a supply of 2.4 V for ECG sampled at 512 Hz.

Lossless-Entropy Encoder with Adaptive Predictor The system in [66] presents a unique lossless ECG encoder based on an adaptive rending predictor and two-stage entropy encoder. When the design was synthesized in 0.18 μm technology, it consumed 36.4 μW at operating frequency of 100 MHz. It achieved a compression ratio of 2.43×.

5.3 Proposed QRS Detection Architecture

The overall block diagram of the proposed ACLT architecture, along with the compressor, is illustrated in Fig. 5.2. In this chapter, the main contribution is in the QRS detection architecture and compressor. Even though the ultimate goal of the compressed data is to be transmitted wirelessly, issues related to the transmitter

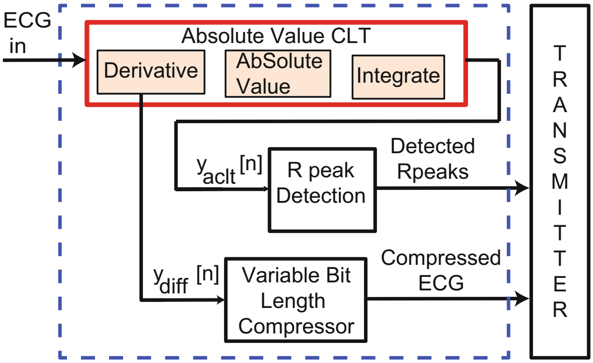


Fig. 5.2 Block diagram of proposed algorithm

such as transmission error are beyond the scope of this chapter. However, in IoT devices, it is necessary to quantify the packet error rate with regard to the signal-to-noise ratio of the wireless transmitter [76].

QRS detectors should be robust enough to deal with the noise and artifacts mentioned in the previous section. It is challenging to come up with a generalized system that deals with all the artifacts at the same time. Filtering has been widely used especially for removing low-frequency noise, baseline drift, and high-frequency interference. Transformation is applied to enhance a portion of the ECG waves. Our proposed system provides optimized QRS detection architectures that could deal with all the artifacts with minimum hardware resources without compromising accuracy.

5.3.1 Algorithm Formulation

Conventional ECG processing flow consists of pre-processing, transformation, and thresholding. Each of these stages requires huge computation in filtering and enhancing ECG. In this proposed technique, the pre-processing and transformation are lumped into one component, forming a modified version of curve length transform (CLT). CLT was reported in [54, 59] and it offers a computationally efficient QRS-detection technique.

CLT, for a discrete signal y_i over a time window ω, is given in Eq. (5.1). Equation (5.1) is referred to as conventional-CLT (C-CLT) in this chapter. The CLT can be re-written and evaluated as in Eq. (5.2). The symbol Δi^2 corresponds to the square of the sampling period (which is a constant value) and replacing it with a nonlinear scaling factor C^2 adds flexibility to manipulate the length-response ratio. C^2 is determined experimentally, taking into account the window size and the maximum height of the QRS complex. By choosing a proper value for it, a particular portion of the signal is improved and boosted in comparison to the rest of the signal.

$$L(\omega, i) = \sum_{i-\omega}^{i} \sqrt{1 + \left(\frac{\Delta y_i}{\Delta i}\right)^2} \Delta t \tag{5.1}$$

$$L(\omega, i) = \sum_{i-\omega}^{i} \sqrt{C^2 + \Delta y_i^2} \tag{5.2}$$

As shown in Eq. (5.2), the CLT integrates successive lengths over a fixed window. Hardware realization of Eq. (5.2) would require addition, multiplication, and calculation of the square root. In order to minimize the resources, Eq. (5.2) could be reformulated as in Eq. (5.3) where the square root is removed. In this chapter, Eq. (5.3) is referred to as squaring-CLT (S-CLT).

$$L(\omega, i) = \sum_{i-\omega}^{i} C^2 + \Delta y_i^2 \tag{5.3}$$

Furthermore, Eq. (5.3) is modified to form Eq. (5.4) where absolute value function replaces the squaring. Hence in this approach both the square and square root functions in Eq. (5.2) are replaced by the absolute value function. This becomes an absolute-value-CLT (ACLT). A multiplying factor 4 is added in Eq. (6.3) to relatively enhance higher ECG slopes and suppress noise which is centered at the baseline. Multiplication by a factor of 4 is implemented as shifting in hardware realization. Using this approach, we are minimizing the resources that would be required to implement the CLT. Its performance and required hardware resources, with respect to other approaches, will be discussed in Sect. 5.3.

$$L(\omega, i) = \sum_{i-\omega}^{i} \left| C^2 + |4 \times \Delta y_i| \right| \tag{5.4}$$

All of the above three approaches (Eqs. (5.2), (5.1), (5.4)) could be applied for QRS detection as the CLT also has an inherent behavior for suppressing the baseline wander of ECG. Based on the above analysis, the CLT could be evaluated using these three approaches, namely: (1) conventional CLT, (2) squaring-CLT (S-CLT), and (3) absolute-value CLT (ACLT). Figure 5.3 shows the transforms for ECG data from MIT-BIH record 112, where the signals have baseline wandering. Though all of the three approaches are feasible, in this chapter, only C-CLT and ACLT are implemented and compared since S-CLT has a large amplitude range about the other two approaches to such a degree that its hardware realization would require more bit width. Also, S-CLT has poor performance in suppressing baseline wander as could be observed in Fig. 5.3, and consequently, its detection accuracy was low. The detailed architecture of the ACLT is presented in the next subsection.

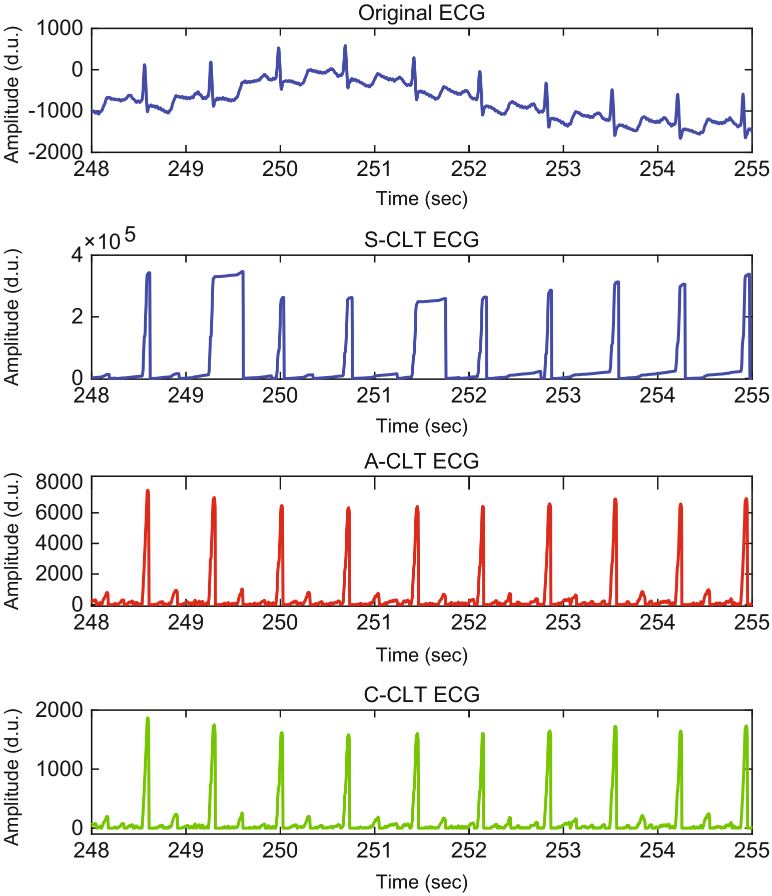


Fig. 5.3 ACLT for MIT-BIH Record 112

5.3.2 *Proposed ACLT Architecture*

Figure 5.4 shows the proposed ACLT architecture for detecting QRS complex. It is an architecture for the algorithm formulated in Eq. (5.4). It performs transformation followed by QRS peak detection using adaptive threshold. The transformation is done using derivative, absolute value, and integration (all lumped into one realization of the ACLT). The transformation distinctively enhances QRS complex even for noisy ECG signals corrupted with baseline wander. Its uniquely inherent behavior removes the need for additional complicated circuits for high-pass or low-pass filters. All of the computations for the transformation are performed using addition and shifting. Moreover, comparison is required for detecting QRS peaks using thresholds. There is no need for multiplication, division, or square root function. Hence its hardware implementation requires only adders, shifters, and

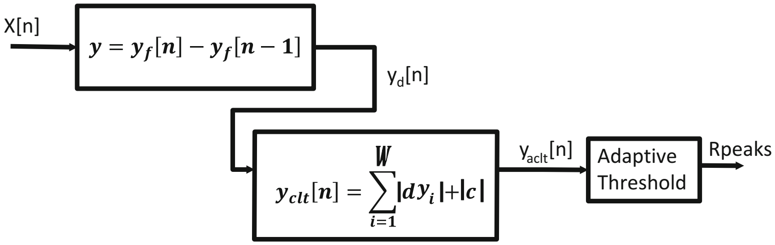


Fig. 5.4 Proposed absolute-value-CLT

comparators. These components are less hardware intensive relative to multipliers, dividers, and square root functions. For instance, if we compare an N-bit multiplier with an N-bit adder, an N-bit multiplier would require N times N-bit adders. Alternatively, a multiplier would need N-times clock cycles. Division and square root are much more complicated than addition or shifting.

The integration over a window in the proposed architecture is pipelined. Pipelining enables it to transform directly whenever there is a new ECG sample. Accordingly, the required clock frequency for the architecture is equal to the sampling frequency of the incoming ECG signal. The sampling frequency of the system is 250 Hz. This is the lowest operating frequency possible for such a configuration. Such a low operating frequency reduces the dynamic power dissipation. Depending on the proposed architecture duty cycling would not give advantage since the design is operating at the sampling rate of the incoming ECG signal. Buffering the ECG signal and then processing at higher frequency would require buffers (SRAM) which add more leakage to the design.

5.3.3 QRS Peak Detection

QRS detection is performed using adaptive threshold. Applying threshold has been commonly used in detecting QRS peaks. However, it is necessary to construct an optimized technique to evaluate the thresholds. A threshold technique where the threshold is set to a mean of all previously detected R_{peaks} is reported in [64]. This threshold is updated according to Eq. (5.5) with every new sample, where the threshold factor P_{Th} is given by Eq. (5.6). The previous threshold is multiplied by a factor with every new sample. Even though using this adaptive threshold produced sensitivity and predictivity above 99%, it requires multiplication with every sample.

$$Th_n = Th_{n-1} * e^{\frac{-P_{Th}}{fs}} \tag{5.5}$$

$$P_{Th} = \frac{0.7 * Fs}{128} + 4.7 \tag{5.6}$$

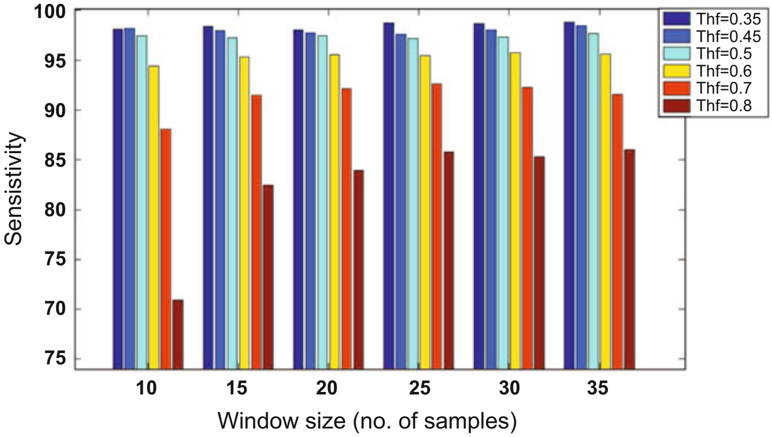


Fig. 5.5 Window and threshold factor selection

In our proposed architecture, the threshold is evaluated based on the equation given in Eq. (5.7). The threshold is updated whenever a new beat is detected and is proportional to the mean of the previously detected QRS peaks. Only eight previously detected QRS peaks are utilized in this stage. In hardware realization, division by 8 is implemented using shifting. The most challenging part in this step is finding the appropriate threshold factor to handle wide morphologically variant ECG waves from different standard databases. Many experiments were done using the standard database from Physionet in order to obtain optimum threshold factor. Figure 5.5 shows the effect of threshold factor on the sensitivity of QRS detection. The experiment was done on MIT-BIH. It is observed that, for a fixed window size of the ACLT, the sensitivity improves with as the threshold factor decreases. Further reduction of the threshold factor would lead to misdetection in which noise or T wave of ECG would be detected as QRS peaks. Figure 5.6 demonstrates the ACLT, along with the threshold, for record 112 from MIT-BIH ECG database.

$$Th_i = Th_{factor} * mean \sum_{k=i-8}^{i} Rpeaks_k \tag{5.7}$$

Once a threshold is defined, the next step is to find a peak in the ACLT signal within a window in which the signal is greater than the threshold. Figure 5.7 shows the FSM that is developed to detect the QRS peaks. State 1 checks if the incoming ACLT signal is greater than a pre-calculated threshold. Initially, the threshold is set to half of the first maximum value of the first 2 s of ECG data. Then the threshold is updated by accumulating newly detected beats, as discussed above, according to Eq. (5.7). When the ACLT signal crosses the threshold value, the system goes on to state 2. In state 2, the system finds the maximum values in a window where the signal is greater than the threshold value. This max value is set as the location of

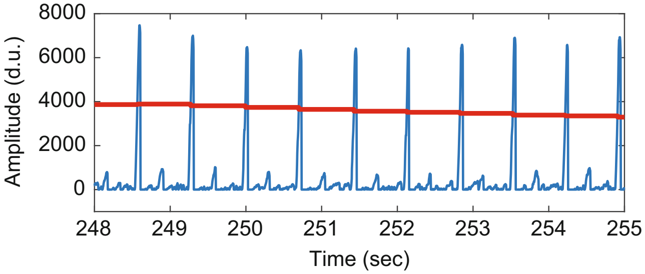


Fig. 5.6 Threshold value for record MIT-BIH 112

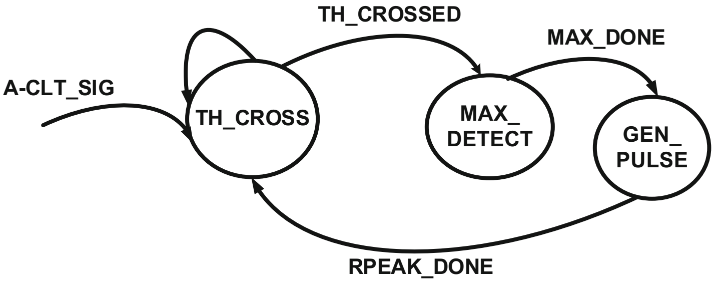


Fig. 5.7 QRS detection FSM

the QRS peak. State 3 generates a pulse indicating the detection of a new beat. This pulse is a fixed offset from the max value obtained in state 2 since the system has to check the whole window for locating the max value. After this, the system goes back to detecting the threshold crossing.

5.3.4 Optimization Parameters

According to the proposed architecture, there are two parameters that need optimum selection. These are the window size (w in Eq. (5.4)) and the threshold factor (Th_{factor}) in Eq. (5.7). In order to set these parameters, the sensitivity and predictivity of the resulting QRS detection are evaluated. Figure 5.5 shows the effect of window size and the threshold factor on the sensitivity of QRS detection. Note that (for a fixed window) the threshold factor has a major impact on Se. For a fixed threshold less than 0.6, the window size does not have much impact on Se. Based on this analysis, a window size of 15 and a threshold factor 0.375 are chosen. Threshold factor 0.375 is $1/2 + 1/8$, so in hardware realization, multiplication by 0.375 is implemented by shifting.

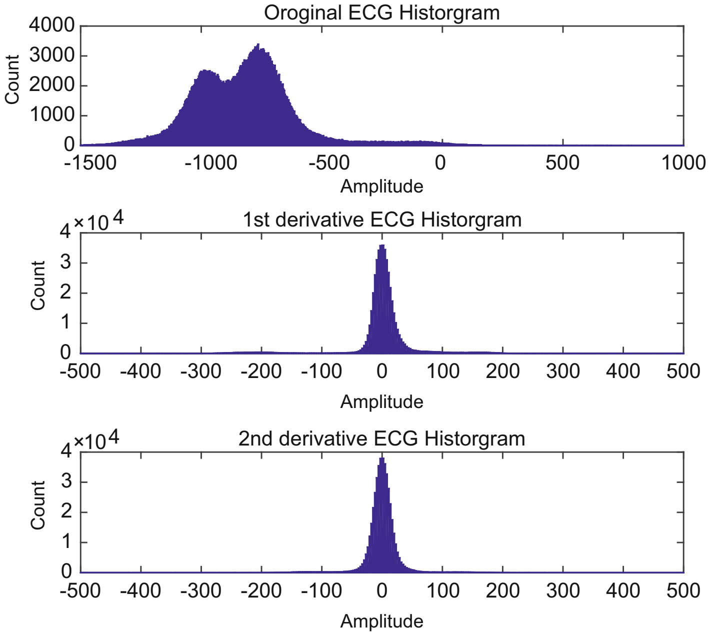


Fig. 5.8 Histogram for data distribution MIT-BIH rec1112

5.4 Proposed ECG Compression Architecture

A novel compression technique based on derivative is proposed. The system takes the first derivative and does a variable bit length compression on the $first\ derivative$ signal. The reason the $first\ derivative$ was chosen is that values from $first\ derivative$ as well as from $second\ derivative$ are concentrated around zero, as shown in Fig. 5.8. However, the amplitude of the original ECG is large amplitudes due to the fact that the QRS complex and its values are concentrated around the baseline. As a consequence more bits would be required to represent the original ECG than were necessary for the $first\ derivative$.

Our objective is to design an ultra-low power compressor that requires minimum hardware resource. The $first\ derivative$ requires only adders. Moreover, the variable bit length encoder requires comparators or a priority encoder which could be easily implemented using combinational logic. Figure 5.9 shows the compressor architecture. However, the $first\ derivative$ would be shared with the ACLT. There will be no additional hardware required to compute the $first\ derivative$. Figure 5.10 illustrates the flow chart for variable length encoder. A lesser number of bits are used for low-amplitude signals, and greater number of bits are used for large amplitude signals. Such encoding reduces the total number of bits required to represent the whole ECG signal, since the $first\ derivative$ values are concentrated around zero.

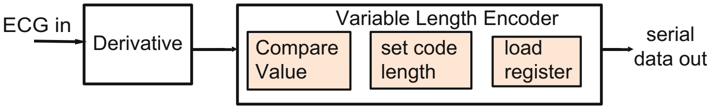


Fig. 5.9 Proposed compressor architecture

5.5 Performance and Results

To evaluate the performance of the algorithms, manually annotated ECG signals from Physionet MIT-BIH Arrhythmia Database and QT database are used [77]. MIT-BIH database contains 30-min-long, 48-two lead-ECG records sampled at 360 Hz, while the QTDB contains 15-min-long, 105-records, out of which 75 contain annotations for the QRS peaks. QTDB contains a wide variation of ECG data collected from other databases [78]. The proposed system was evaluated using the 48 records from MIT-BIH and 75 records from QTDB. MIT-BIH database contains randomly selected subjects as well as subjects with known arrhythmia that have clinical significance [79]. Moreover, the subjects are both men and women aged between 22 and 89 years. It has been widely used as a standard database for evaluating ECG QRS/arrhythmia detectors.

5.5.1 QRS Detection Performance

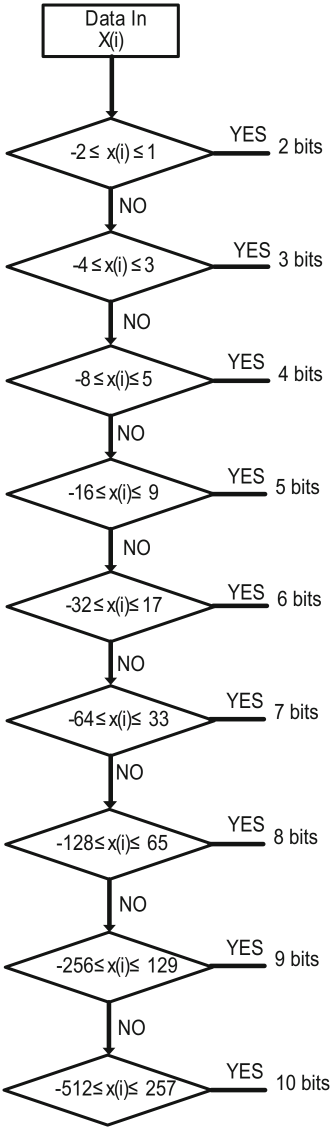
The proposed QRS detection architecture could detect various ECG morphologies including those with baseline wander, motion artifact, and noise corruption. Figure 5.11 shows ECG record 112 from MIT-BIH annotated with reference annotations (green) and detected annotations (red).

The performance of QRS complex detectors is evaluated before it is used in medical devices. The performance metric used in standard procedures is the sensitivity (Se) and positive predictivity (P^+). Detected QRS peaks are compared with reference annotation from experts. The sensitivity and positive predictivity are defined by Eqs. (5.8) and 5.9, respectively, where TP stands for the number of truly detected beats, FN denotes the number of false negative detection in which a beat exists but is not detected, and FP refers to the number of false-positive detection in which a beat does not exist but is detected.

$$Se = \frac{TP}{TP + FN} \times 100 \tag{5.8}$$

$$P^+ = \frac{TP}{TP + FP} \times 100 \tag{5.9}$$

Fig. 5.10 Variable length compressor flow chart



The detection performance obtained by the self-adaptive QRS detectors implemented in this work and other published detectors including [25, 27, 28, 73] and [64] are displayed in Table 5.1. The overall sensitivity of the implemented QRS detectors (based on C-CLT and ACLT) is found to be at levels of 99.0% and 99.37%, respectively. Following the same order, the positive predictivity is 99.3% and 99.38% when evaluated against the annotated beats in MIT-BIH. Table 5.1 shows that proposed ACLT performs well in the order of greater than 99.3% though

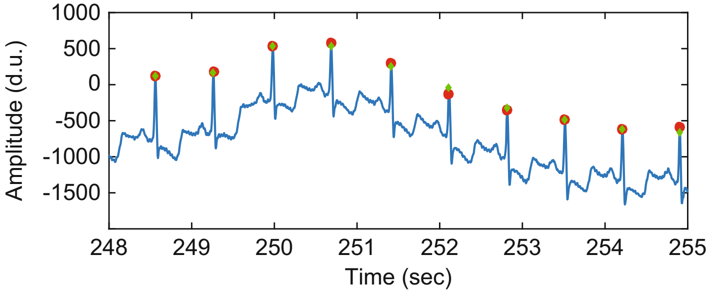


Fig. 5.11 QRS detection for MIT-BIH record 112

Table 5.1 Sensitivity and positive predictivity of QRS complex detectors (MIT-BIH)

Technique	*Se*	P^+
[25]	99.69%	99.77%
[27]	99.8%	99.86%
[28]	99.63%	99.89%
[73]	99.31%	99.7%
[64]	99.54%	99.74%
C-CLT	99.0 %	99.33 %
ACLT	99.37%	99.38%

its implementation is much less complex than that of the other referenced systems. Systems reported in [27], [28], and [73] are based on wavelet transform that requires multiscale decomposition which is implemented using FIR filters [64].

5.5.2 Computational Complexity of QRS Detector

Computational complexity gives a measure to evaluate the system for its suitability in ultra-low power IoT systems. For comparison, we have implemented three versions of the CLT. Moreover, we have made a comparison with the system implemented in [64] along with PAT as implemented in [64]. PAT is a widely reported QRS detection technique.

The computational complexity of the proposed algorithm is measured using the number of multipliers, adders, and comparators needed for the design. Table 5.2 reveals the resources required for the proposed architecture. The main superiority of the proposed ACLT architecture is that it does not require any multipliers. Though the number of adders and additions per second required in the proposed system is greater than in [64], the total operations per second is less than 50%. The proposed system requires 35% of the comparators required in [64] and 53% of the PAT implemented in [64].

Table 5.2 Resource consumption

Technique	[64]	PAT as in [64]	Conventional CLT	Proposed ACLT
Memory cells	28	123	18	18
Multipliers	6	6	1	0
Adders	5	41	13	13
Comparators	–	–	3	3
Square root	–	–	1	–
Square root./s	–	–	250	0
Mult./s	1107	1201	250	0
Adds./s	1205	1107	1261	1261
Comp./s	2163	1416	750	750
Total Ops./s	4475	5434	2512	2012

Table 5.3 Hardware resources and power

Technique	Conventional CLT	Proposed ACLT
Combinatorial cells	1082	657
Sequential cells	341	445
Buffers/inverters	101	146
Total cells	1423	1102
Area μm^2	13,940	10,074
Operating frequency	250 Hz	250 Hz
Leakage power	7.3 nW	5.16 nW
Dynamic power	1.6 nW	1.34 nW
Power	8.9 nW	6.5 nW

Relative to the C-CLT implemented by the authors, the proposed system does not use squaring or square root functions. Both squaring and square root functions (especially square root) are hardware-intensive operations. This implies that there is a 100% saving in multiplications and square root by implementing ACLT. Moreover, there is a saving of 27% on the power consumption as revealed in Table 5.3. Even though these operations are removed in order to attain ACLT, the performance is comparable. Even if we compare their sensitivities, ACLT achieves better results. If we look at Table 5.1, the proposed ACLT has a sensitivity and predictivity greater than 99%.

5.5.3 Compression Architecture Performance

Our proposed compressor is based on variable bit length for the first derivative of an ECG signal. Figure 5.12 illustrates a sample ECG and its first derivative. Relative to the original ECG, the signal amplitude range is reduced by a factor of 2. In addition,

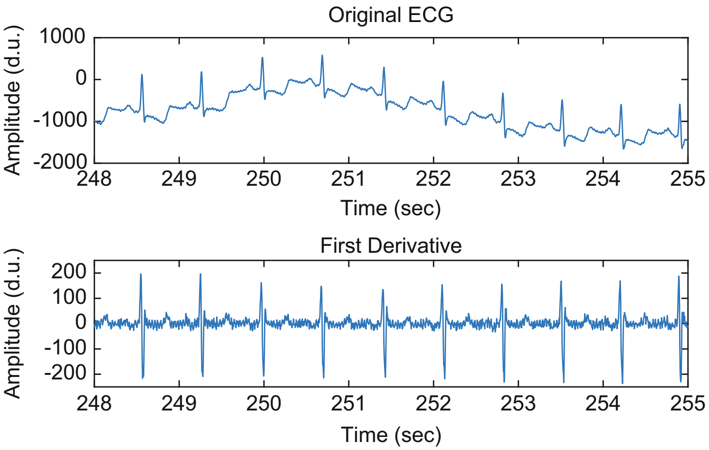


Fig. 5.12 Compression: first derivative for MIT-BIH record 112

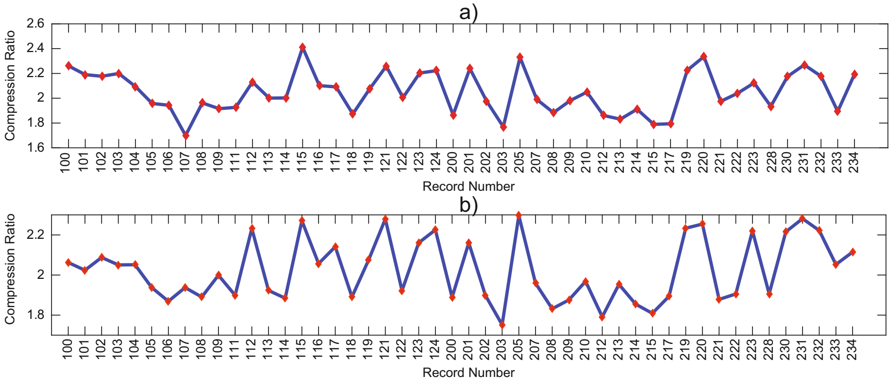


Fig. 5.13 Compression ratio for MIT-BIH (**a**) using first derivative and (**b**) using second derivative

the values in the first derivative are concentrated around zero; though, the original ECG has baseline drift.

The bit compression ratio is evaluated as in Eq. (5.10) where a total number of bits of uncompressed samples corresponds to the product of the number of samples with a fixed number of bits per sample (Eq. (5.11)). The MIT-BIH is sampled using 11 bits/sample. The number of bits of the compressed data corresponds to the summation of all bits from each sample (Eq. (5.12)). An average compression ratio of 2.05× and 2.10× is attained using the first and second derivative of ECG from MIT-BIH (as illustrated in Fig. 5.13). Figure 5.13 presents the compression ratio for all the records from MIT-BIH database. The compression ratio for all records is illustrated because the records have different morphologies and represent various cardiac conditions. Hence, it is a verification that the compression algorithm could handle various morphologies at a small range of compression ratio (within 1.7× and 2.4×).

$$\text{BCR} = \frac{\text{T. No. of bits uncompressed samples}}{\text{T. No. of bits compressed samples}} \tag{5.10}$$

$$\begin{aligned}&\text{Total number of bits uncompressed samples}\\&\quad = \text{No. samples} \times \text{(bits/sample)}\end{aligned} \tag{5.11}$$

$$\begin{aligned}&\text{Total number of bits compressed samples}\\&\quad = \sum \text{All bits of each sample}\end{aligned} \tag{5.12}$$

5.5.4 Hardware Implementations and Synthesis Results

The proposed architecture is coded using Verilog and simulated for functional verification. Its realization schematic is shown in Fig. 5.14. The design was synthesized using state-of-the-art tools from Synopsys, and layout was also generated. The standard cell library was fully characterized in silicon and is in an industry-standard tape-out-ready form. The standard cells were three flavors: LVT, RVT, and HVT. Though LVT cells have high leakage, they are more suitable for high speed applications and HVT cells for low-leakage applications where speed is not a major concern. RVT lies between LVT and HVT in terms of leakage and speed. The implementation was done using HVT cells, as HVT cells have more than 10× lower leakage than RVT cells in 65 nm, in addition to the design being operated at low frequency. Post-layout power analysis shows that the ACLT system consumed a total power of 6.5 nW when operated from a supply of 1 V at an operating frequency of 250 Hz. The leakage power is 5.16 nW, accounting for 79% of the total power. The leakage power could be optimized by powering from a lower supply voltage, and the system could go up to 0.4 V for 65 nm technology [52]. We can estimate leakage saving at lower voltages, as the leakage is linearly related to the supply voltage. For instance, if the leakage at 1 V is 5.16 nW, then the leakage will be 2.064 nW at 0.4 V (which is a reduction of 60% in the leakage power).

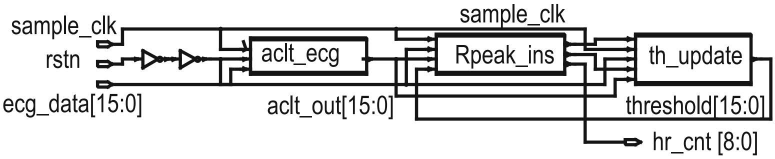


Fig. 5.14 Schematic ACLT core

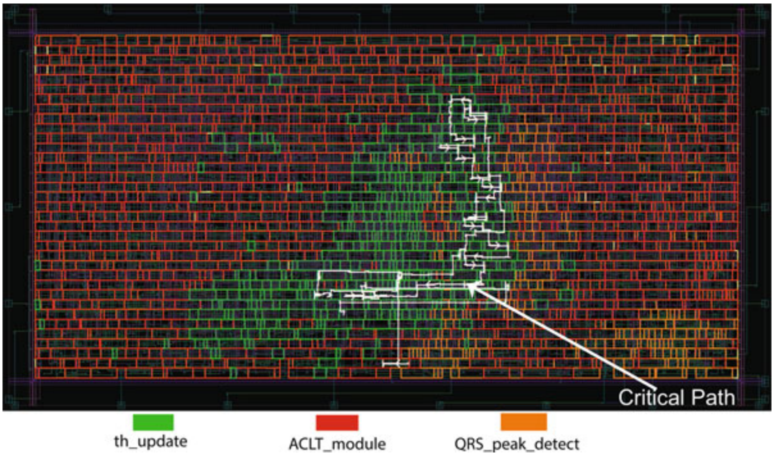


Fig. 5.15 Layout of the ACLT core

The layout of the proposed ACLT architecture is revealed in Fig. 5.15 which was generated using IC compiler from Synopsys. Design hierarchy and the worst case timing path are annotated in the figure. Timing verification was also performed, and the design has positive slack meeting all setup and hold time requirements. Timing closure was achieved using design constraint based on the standard cell characteristics.

5.6 Compressor Comparison with Literature

Table 5.4 shows the comparison of the proposed compressor with literature. The proposed lossless compression architecture consumed only 3.9 nW when operating at a frequency of 3 kHz, at supply voltage 1 V. The leakage is 0.51 nW, accounting for 13.1%. Operating frequency is set to 3.0 kHz so as to transmit the maximum number of bits serially from the variable length encoder within the sampling time of the input ECG signal. Even though the proposed architecture has a compression of 2.05, slightly lower than that reported in [65, 66], its implementation only requires 0.179 K gates and only consumes 3.9 nW. The system in [66] (being a standalone compressor) consists of a predictor followed by the entropy encoder. However, the compressor in [65] is part of a complete ECG processing system that includes an analog front end. Since the compressor subsystem performance (power and area) are reported separately, these metrics are used for comparison. Therefore, the comparison that is reported in Table 5.4 is apple-to-apple.

Table 5.4 Compressor comparison with published work

	[66]	[65]	Proposed
Technology (nm)	180	350	65
Oper. freq.	100 MHz	32 kHz	3 kHz
Supply voltage	1.8	2.4	1
Compression ratio	2.43	2.25	2.05
ECG channels	1	1	1
Total gate count	3.57 K	2.26 K	0.179 K
Power (μW)	36.4	0.535	0.0039

5.7 Summary

This chapter presented a real-time QRS detector and ECG compression architecture for energy constrained IoT healthcare wearable devices. An ACLT that effectively enhances QRS complex detection with minimized hardware resources was proposed. The proposed implementation required adders, shifters, and comparators and avoided the need for any multipliers. QRS detection was accomplished using adaptive thresholds in the ACLT transformed ECG signal. The proposed QRS detector achieved a sensitivity of 99.37% and a predictivity of 99.38% when validated using databases acquired from MIT Physionet. Furthermore, a lossless compression technique was incorporated into the proposed architecture using the ECG signal first derivative and variable bit length, an average compression ratio of 2.05 was achieved when evaluated using MIT-BIH database. The proposed QRS detection architecture was implemented using 65 nm low-power process; it consumed an ultra-low power of 6.5 nW when operated at a supply of 1 V. Also, the proposed compressor consumed only 3.9 nW when operated at a supply of 1 V.