# A novel VLSI implementation of lossless ECG data compression technique using intelligent Slope Predictor and Modified Huffman Coding

V. R. S. RAJESH KUMAR, A. SIVANANTHA RAJA[a]
*Research Scholar, Anna University, Chennai, Tamilnadu, India*
*[a]Associate Professor of ECE, Alagappa Chettiar College of Engineering and Technology, Karaikudi, India*

In medical applications, many techniques have been proposed for lossless compression of ECG signals. However, due to more complex and less compression ratio, those techniques cannot be suited to transmit ECG signals. A novel VLSI implementation of lossless ECG data compression technique is proposed to save power, area and to get better compression ratio (CR). The proposed algorithm consists of an intelligent Slope Predictor (i-SP) algorithm for lossless ECG data compression, and with Modified Huffman Coding (MHC) for an entropy encoder. The i-SP algorithm segments the raw ECG data in both QRS and smooth sections. Then Modified Huffman Coding (MHC) encodes the predicted differences of ECG data. The VLSI architecture of the proposed technique contains 2.1k counts of logic gates, its synthesized area 24.6k$\mu m^2$ and power 22.5$\mu W$ in 180nm CMOS technology. From arrhythmia database of the MIT-BIH, the proposed technique achieves a CR of 2.87. As compared with the existing lossless ECG data compression techniques, the proposed system achieves high CR, low area and less power consumption.

(Received May 1, 2015; accepted May 7, 2015)

*Keywords*: ECG data compression, Intelligent slope predictor, Modified Huffman Coding

## 1. Introduction

The ECG [1] is a realistic graphic display of the electrical action of the heart. Because of minimal effort and non-invasion, ECG signal has been reached out for heart disease diagnosis and mobile check-in. For storage and transmission of huge signal information, it is important to compress the ECG signal data.

To get critical signal compression, lossy compression is desirable over a lossless compression. Lossless compression is only an accurate reconstruction of the original signal and is in light of the thought of breaking a data into a "smaller" structure for transmission or storage capacity and then assembling it back on the other end so it can be utilized once more.

Lossless ECG compression methods are naturally classified into three main categories:

1. Data compression techniques: Amplitude Zone Time Epoch Coding (AZTEC) [2], Scan-Along Polygonal Approximation (SAPA), Coordinate Reduction Time Encoding System (CORTES), Differential Pulse Code Modulation (DPCM), Turning Point (TP) [3] and so on.
2. Transform coding techniques: Karhunen-Love Transform (KLT), Discrete Cosine Transform (DCT), Walsh Transform (WlshT), Wavelet Transform (WT) and so on.
3. Parameter extraction techniques: Peak-Peaking (PP), Cycle-Pool-Based Compression (CPBC), linear prediction (LP), neural network (NN) and so on.

As of late, wavelets are broadly utilized for both one dimensional and two dimensional ECG compressions [4-7]. The majority of the papers demonstrated great compression performances for customary ECG cases. Among transformation plans, Wavelet Transform (WT) [8-12] has developed into exceptionally famous because of the way that the time-frequency piece for the WT-based technique can better limit the signal components in time-frequency space. But WT, Orthogonal Transform (OT) [13] and Discrete Cosine Transform (DCT) [14] have additionally been utilized for getting compressed ECG data. These strategies are principally in view of linear prediction and long-term prediction techniques.

Data compression [15-16] can be utilized to diminish transmission energy as well as transmits RF module information with limited bandwidth. However, this technique is a lossy compression technique. So, this technique is not suitable for medical monitoring applications. A slope prediction [17] had been utilized to estimate the pattern of the ECG signals. An adaptive trending prediction (ATP) [18] and fuzzy decision [19] techniques with two-stage Huffman entropy coding has been developed in VLSI implementation for low complex lossless compression methods. However, the lossless compression techniques [15-19] are presented high CR for biomedical applications. Due to more complexities, these techniques are not suitable for VLSI implementation.

This paper proposes an intelligent slope predictor (i-SP) algorithm for lossless ECG data compression. This algorithm segments the raw ECG data into QRS section and smooth section. Later, the first order linear predictor is applied to QRS section and the $0^{th}$ order linear predictor is applied to the smooth section. These predicted differences is further encoded using Modified Huffman Coding

(MHC) which has a lesser range for faster encoding with lesser computation than the conventional technique.

The paper is organized as follows. Section II describes the fundamental ECG waveform; Section III demonstrates the proposed method; Section IV discusses compression and distortion measures and Section V concludes the paper.

## 2. Fundamental ECG waveform

ECG represents different electrical phases of a cardiac cycle. From cardiac cells, it represents the space of the movement activity. ECG gives a measure of the electrical currents started in the additional cellular fluid due to the potential changes over the cell membrane. A normal scalar ECG lead is demonstrated in Fig. 1, which goes through the heart's walls, bringing on synchronized contractions. The waves named by Willem Einthoven, who was granted the 1924 Nobel Prize in Medicine for his spearheading effort on ECG.

It comprises of a position of progressive waves indicated by $P$, $Q$, $R$, $S$, $T$ and $U$ waves. Investigation of every part of the waveform is imperative to comprehend and later misuse the redundancies in the signal. The $P$ segment shows the period when the atria are electrically fortified to pump the blood into the ventricles. The $QRS$ segment is a complex section which demonstrates that the ventricles are electrically fortified to pump the blood out. The $ST$ segment demonstrates the measure of time from the end of the constriction of the ventricles to the start of the $T$ wave while the $T$ wave is the recuperation time of the ventricles. The last partition, the $U$ wave is the re-polarization of the papillary muscle strength yet it is infrequently seen.

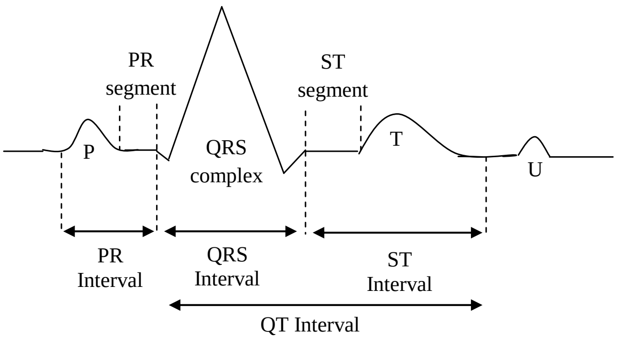



*Fig. 1. Fundamental ECG waveform.*

To perform the diagnosis, the waveform of the ECG signal and additionally the intervals between the various sections of the ECG waveform that are delivered, owing to the way that they give data about the coordination between the diverse occasions amid a cardiac cycle. The typical duration of a portion of the ECG parts in grown-up patients is:

- P wave: less than 120ms.
- PR interval: in between 120ms and 200ms.
- QRS complex: also less than 120ms.
- QT interval: less than 460ms.

There are noteworthy varieties in the ECG relying upon the individual or even its own condition.

- P wave-0.25 mV.
- R wave-1.60 mV.
- Q wave-25%R.
- T wave-0.1mV-0.5 mV.

## 3. Proposed method

From all kind of compression techniques, lossless compression is the most essential feature for ECG signals. Lossless ECG compression is necessary for storage and transmission of ECGs. According to the regulations of law, medical signals after lossy compression technique cannot be utilized as a part of the diagnostics in numerous countries. As a medical diagnosis may rely on the specialist's elucidation of the medical signals sent by a patient, the dependability of the received signals must be guaranteed. Here is the place lossless compression gets to be essential, so as its name demonstrates, there is no loss of data. And for instance, lossy compression cannot be guaranteed. Moreover, in few countries it is prohibited by law to lossy compress images utilized for medical diagnosis.

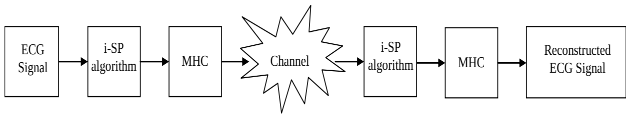



*Fig. 2. Block diagram of a lossless coder/decoder system.*

As per Fig. 2, an intelligent Slope Predictor (i-SP) algorithm segments the raw ECG data in QRS section and smooth section. Later, the first order linear predictor will be applied to QRS section and the $0^{th}$ order linear predictor will be applied to the smooth section. These predicted differences will be further encoded using Modified Huffman Coding (MHC) which has a faster encoding with lesser computation than the conventional technique. The output information of the decoder gets accurate data of the signal before being compressed, so that the data is conserved in its unique form.

### 3.1 Intelligent Slope Predictor Algorithm

Predictive coding [20] has been utilized effectively for waveform compression, for example, audio coding. In predictive coding, both the transmitter and receiver predict the value of the recent sample $\mathrm{x(n)}$ based on the samples that have already been transmitted and received. A slope based predictor to estimate the value of the recent ECG $(x(n))$ sample based on previous 2 samples $(x(n-1), x(n-2))$, i.e., it computes the current sample value using the same signal slope obtained from previous two samples. If the predicted value is $\tilde{\mathrm{x}}(\mathrm{n})$, then itself the prediction error needs to be transmitted to allocate a perfect reconstruction of the current sample. This predicted value is subtracted from the recent value to obtain the prediction error of the current sample. The two predictors, used more frequently, the first-order predictor, estimation for slope predictor and prediction error are given by (1), (2) and (3).

$$first\ order\ predictor\ x(n) = x(n-1) + e(n) \quad (1)$$
$$\tilde{x}(n) = 2 * x(n-1) - x(n-2) + e(n) \quad (2)$$
$$e(n) = x(n) - \tilde{x}(n) \quad (3)$$

Where $\tilde{x}(n)$ is prediction estimate, $e(n)$ is the prediction error. The dynamic range of $e(n)$ is generally much smaller than that of the ECG signal. The second-order estimated value for the next sample would be modified by the same difference as the previous value. This process works exactly for a signal with a constant slope. Noted that, for achieving lossless compression it needs maximum $(L+2)$ bits to fully represent $e(n)$, where L is the bit-width of $x(n)$. With the proposed scheme, only prediction error needs to be transmitted instead of original ECG samples. At the receiver side, the reverse process is done to reconstruct the original data samples.

### 3.2 Modified Huffman Coding (MHC) technique

Huffman coding [21] is one of the most established and most famous techniques for data compression. It is in view of that the estimations of an information stream are not probable, so every stream contains a high frequency of specific characters, while others are not all that regular.

As all the measurable techniques do, Huffman coding creates variable-size codes. The length of the allocated code to every symbol relies on its frequency of appearance. Subsequently, the shorter codes are allocated to the symbols that seem all the more habitually. In addition, Huffman coding fits in with the group of prefix codes, so no symbol is a prefix of some other symbol. That is imperative as every symbol cannot be divided from the lay relying upon its length. For instance, if $A$ switches to 1, $B$ switches to 01 and $C$ switches to 101, the decoder will be not able to separate the symbols as the symbol $A$ ensues to be a prefix of another symbol. In the event that the decoder got the succession 101, it would be not able to figure out whether it was an $A$ took after by a $B$ or simply a $C$.

The methodology of coding is indicated by a case. To acquire the Huffman code given five symbols $a1$, $a2$, $a3$, $a4$, $a5$ with probabilities $P(a1) = 0.4$, $P(a2) = 0.2$, $P(a3) = 0.2$, $P(a4) = 0.1$, $P(a5) = 0.1$, the following steps must be taken after: once the probability of each symbol is ascertained, a binary tree is built and it will be responsible for giving the last coding. The production of that tree is executed as it takes over:

1. Make a sorted list in descending order of every last one of probabilities.
2. The two components with the minimum probabilities are chosen and another component is made. Its probability is the expansion of both probabilities.
3. Realign the probabilities list with the new arrangement of components.
4. Until a single node is obtained repeat the step 2 and step 3.

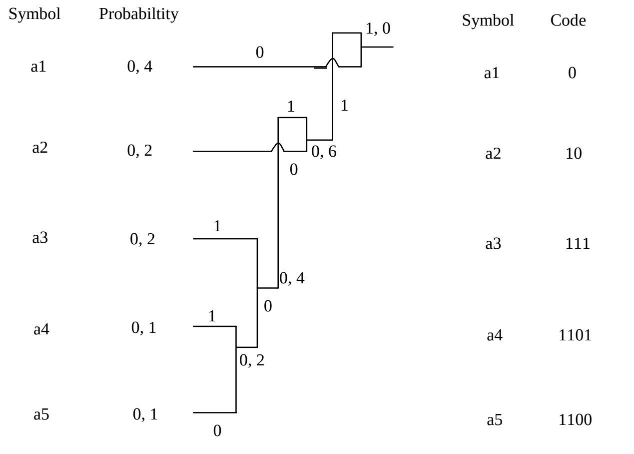



*Fig. 3. Huffman Codes.*

Through the making of the binary tree it is conceivable to allocate a binary code to every component of the alphabet. The process of Huffman coding is shown in the Fig. 3. When the tree is finished, it shows the task of bits.

The compression accomplished by the Huffman codification relies on the circulation of the source components. There is a situated of 5 components, so 3 bits were required to codify them. By utilizing the Huffman coding the normal length can be ascertained as:

$$E[l] = \sum_{i=1}^{n} l_i P_i \quad (4)$$

Where $n$ the length of the alphabet is, $l_i$ is the length of the Huffman coding for each element, and $P_i$ is its probability. In the illustration over, this normal length is

$$E[l] = 0.4 \times 1 + 0.2 \times 2 + 0.2 \times 3 + 0.1 \times 4 + 0.1 \times 4 = 2.2\ bits/symbol$$

Since there were additional two symbols with the same probabilities, the procedure of coding is not inimitable. Nevertheless, what can be guaranteed is that the normal output size will be the same.

The reconstruction procedure of the code is acknowledged by covering the binary tree to the terminal node. This is conceivable as the Huffman code and it has the property of being quick, so that the decoder dependably knows when the coder procedure is done. Besides Huffman coding is a prefix code, it is important to transmit or store the binary tree to decipher the information.

The Huffman coding does not lose information. While transmission or storage, it gets an error which can influence a single bit, this is deciphered into more than one error through the reconstruction. Generally, some kind of defense for the information after the coding stage is utilized.

Huffman coding is consolidated with other compression methods and it is in text compression as well as in images or video.

### *3.2.1 Algorithm*

**Input:** The values $f_i, i = 0, \dots, n-1$ of a Huffman tree $T$ with height $h$ and a bit stream $b_j,\ j = 1, \dots, N, j$ is the final decoded bit, $q$ is a stream of 1-bit.

**Output:** The Binary coding tree symbols $b_k, k = 1, \dots, M$ corresponding to the bit stream of input. $k$ is the final decoded symbol

$j \leftarrow 0, k \leftarrow 0$
$f_{-1} \leftarrow 0$
**While** $j < N$ **do**
**if** $(h < N - j)$ **then**
$d \leftarrow b[j + 1 : j + h]$
**else**
$d \leftarrow b[j + 1 : N]\ \#\ q_{h-N+j}$
**endif**
$binsearch(f, d + 1, index)$
// The $d + 1$ decides the $binsearch.\ index$ locates its location.
$k \leftarrow k + 1$
$b_k \leftarrow f_{index}$
$j \leftarrow j + h - \log_2(f_{index} - f_{index-1})$
**Enddo**

In decoding final symbol of the text, it may not be accessible $h$ bits. In which case $d$ is attained by attaching the bit stream with sufficient number of 1 bit. In decoding bit design, it does a binary search among $n$ symbols that obliges $O(log\ n)$ correlations.

On the off chance that there are $m$ code words, our algorithm will run in $O(m\ log\ n)$ times. It might be said here that keeping in mind the end goal to enhance repeated application of Huffman coding it is critical to have the overhead in place of a Huffman tree as effectively as could be allowed. So, such effective representation of a Huffman header as proposed in this paper offers ascend to the chance of applying repeated Huffman coding in situations where further compression is attractive.

## 4. Result and discussion

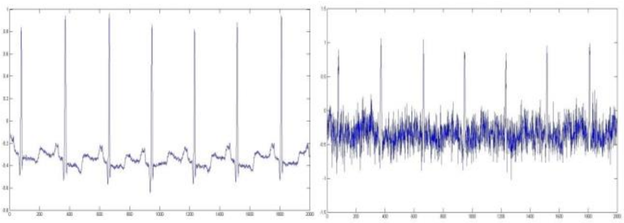
By eliminating the conceivable redundancy by increasing the compression ratio, all data compression algorithms try to minimize data storage. A high compression ratio is ordinarily wanted; however, utilizing this alone to analyze data compression algorithms is not adequate. In addition, the transmission capacity, sampling frequency and exactness of the original information will influence the compression ratio. A data compression algorithm should represent the information with satisfactory fidelity. In ECG signal compression, the clinical agreeability of the reconstructed signal must be decided through visual examination by cardiologists. The error signal resulting between the reconstructed signal and the original signal may additionally be measured numerically. A lossless compression data algorithm creates zero residual and the reconstructed signal precisely represents the original signal. In any case, clinically adequate quality is neither ensured by a low non zero residual nor precluded by a high numerical residual. The criteria for testing compression execution of an algorithm incorporate the following two parts: Compression ratio and Computational complexity. The computational complexity part will be a piece of the pragmatic execution thought.

### 4.1 Compression Ratio (CR)

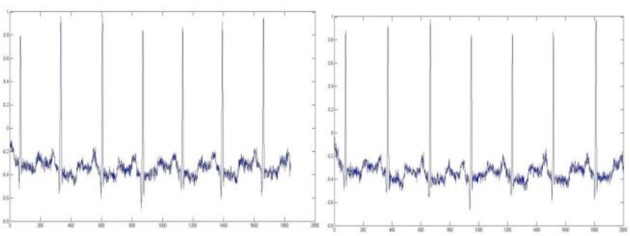
The span of compression is constantly computed by CR, which is portrayed as the proportion between the bit rate of the original signal ($b_{original}$) and the bit rate of the compressed ($b_{compressed}$).

$$CR = \frac{b_{original}}{b_{compressed}} \tag{5}$$

The issue of utilizing the above definition (5) of CR is that each algorithm is nourished with an ECG signal that has an alternate sampling frequency and an alternate number of quantization levels; therefore, the bit rate of the unique signal is not standard. In this section, the execution of proposed compression algorithm is assessed and contrasted with other algorithms. For this reason, information from the MIT-BIH arrhythmia database is utilized to assess the proposed compression algorithm. Fig. 4 is established to allocate visual evaluation of the value of 2000 samples length ECG signal.



a) Original ECG Signal (b) Noisy ECG Signal



(c) Compressed ECG Signal (d) Reconstructed ECG Signal

*Fig. 4. (a) Original ECG Signal, (b) Noisy ECG Signal, (c) Compressed ECG Signal, (d) Reconstructed ECG Signal.*

### 4.2 Implementation results

In order to verify the functions of this design before tape-out, the VLSI implementation of this design is developed in Verilog code, simulated using NC-Verilog tool and synthesized by design compiler in TSMC 180nm CMOS technology.

It consumes 22.5 $\mu W$, contains 2100 gate counts, occupies 24687$\mu m^2$area, and normalized area is 1, shown in Table 1.

*Table 1. Comparison of existing methods and proposed method.*

| | Process (nm) | Operation Frequency (MHz) | Voltage (V) | CR | power ($\mu W$) | Gate Counts | Area ($\mu m^2$) | Normalized Area |
|---|---|---|---|---|---|---|---|---|
| Adaptive Power Conserving (APC) algorithm | 180 | 100 | 1.8 | 1.9 | 150 | 13400 | 134K | 3.9 |
| Discrete Pulse Code Modulation (DPCM) Prediction with Golomb-Rice Entropy | 65 | 24 | 1 | 2.38 | 170 | 53900 | 58K | 12.9 |
| Efficient Micro Control Unit (MCU) with a Reconfigurable Filter | 180 | 100 | 1.8 | 2.38 | 5800 | 3770 | 76K | 2.21 |
| Two Stage Entropy Encoder (TSEE) | 180 | 100 | 1.8 | 2.43 | 36.4 | 3570 | 46K | 1.34 |
| Fuzzy Decision (FD) and Two-Stage Huffman Coding | 180 | 100 | 1.8 | 2.53 | 28.3 | 2780 | 34.4K | 1 |
| i-SP and MHC | 180 | 100 | 1.8 | 2.87 | 22.5 | 2100 | 24.6K | 1 |

From Fig. 5, APC algorithm gets the minimum CR value, which is 1.9. As compared with the APC algorithm, DPCM Prediction with Golomb-Rice Entropy algorithm, MCU with a Reconfigurable Filter technique, TSEE algorithm and FD and Two-Stage Huffman Coding algorithm gets better CR values, which are 2.38, 2.38, 2.43 and 2.53 respectively. However, CR of the proposed method gives better result as compared with all other existing techniques. The proposed technique achieves the maximum CR of 2.87.

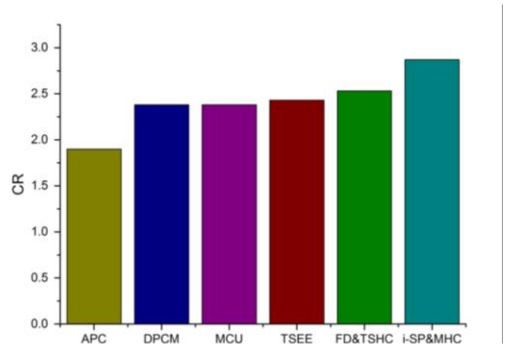



*Fig. 5. Comparison results of CR.*

#### *4.2.1 Power*

From Fig. 6, MCU with a reconfigurable filter technique consumes more power as compared with other techniques, which is 5.8mW. APC algorithm, DPCM Prediction with Golomb-Rice Entropy, TSEE and FD and Two-Stage Huffman coding consumes 150 $\mu W$, 170 $\mu W$, 36.4 $\mu W$ and 28.3 $\mu W$ of power respectively.

However, power of the proposed method gives better result as compared with all other existing techniques. The proposed technique consumes 22.5 $\mu W$.

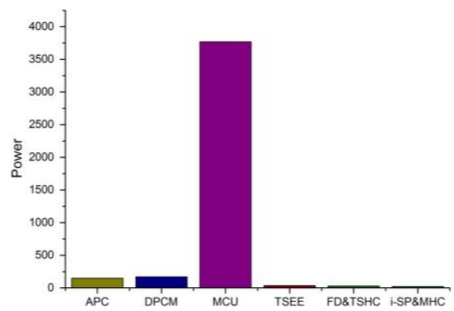



*Fig. 6. Comparison results of Power in $\mu W$.*

#### *4.2.2 Gate Counts*

From Fig. 7, the proposed technique realizes less gate counts compared to other techniques, which counts 2.1k number of gates only. Adaptive power conserving algorithm, DPCM Prediction with Golomb-Rice Entropy, MCU with a reconfigurable filter technique and FD and Two-Stage Huffman coding methods occupied 13.4k, 3.77k, 3.57k, and 2.78k respectively. But the reference TSEE counts 53.9k number of gates, which is worse compared to all other techniques.

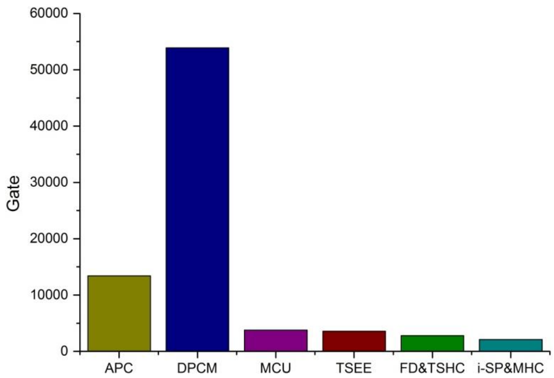



*Fig. 7. Comparison results of Gate counts.*

#### *4.2.3 Area*

From Fig. 8, the area of the proposed method occupies 24.6K$\mu m^2$ only. And the APC algorithm occupies more area as compared with all other ECG compression techniques, which occupies 134 K$\mu m^2$. Area of the DPCM Prediction with Golomb-Rice Entropy, MCU with a reconfigurable filter technique, TSEE and FD and Two-Stage Huffman coding methods occupy 58K $\mu m^2$, 76K$\mu m^2$, 46K$\mu m^2$ and 34.4K$\mu m^2$ respectively.

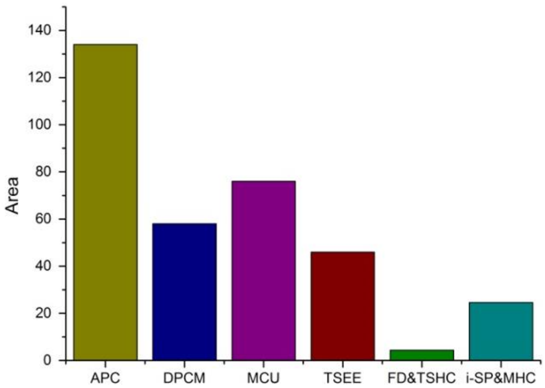



*Fig. 8. Comparison results of Area in K$\mu m^2$.*

#### *4.2.4 Normalized area*

From Fig. 9, the normalized area of the proposed method and FD and Two-Stage Huffman coding method both are same. And DPCM Prediction with Golomb-Rice Entropy is having more normalized area as compared with all other ECG compression techniques, which is 12.9. Other references APC algorithm, MCU with a reconfigurable filter technique and TSEE gets 3.9, 2.21, and 1.34 respectively.

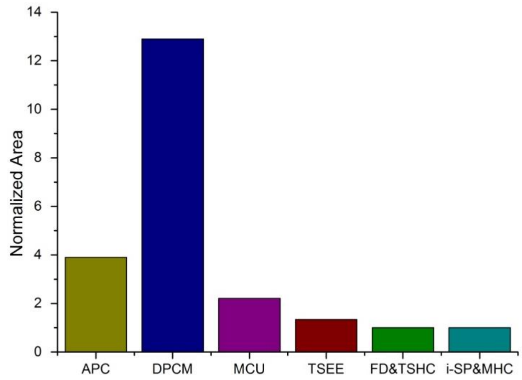



*Fig. 9. Comparison results of Normalized Area.*

## 5. Conclusion

This paper implements a lossless ECG compression algorithm based on intelligent Slope Predictor and Modified Huffman Coding techniques. The i-SP algorithm segmented the raw ECG data into QRS section and smooth section. The differences predicted by i-SP are encoded using MHC, which have lower range for faster encoding with less computation than the conventional technique. From all patterns of the MIT-BIH Arrhythmia database, the proposed lossless ECG data compression algorithm achieves a CR of 2.87, which is much better than other lossless compression algorithms. The simulated power of the proposed method consumed less power as compared to other techniques. The gate counts and area are also less compared to other techniques.

## References

[1] J. H. Hampton, The ECG Made Easy. Churchill Livingstone, Sixth Edition 2003.
[2] J. R. Cox, F. M. Nolle, H. A. Fozzard, G. C. Oliver, IEEE Transactions on Biomedical Engineering, **15**, 128 (1968).
[3] W. C. Mueller, Biomedical Sciences Instrumentation, **14**, 81 (1978).
[4] J. P. Abenstein, W. J. Tompkins, IEEE Transactions on Biomedical Engineering, **29**, 43 (1982).
[5] U. E. Ruttimann, H. V. Pipberger, IEEE Transactions on Biomedical Engineering, **26**, 613 (1979).
[6] S. K. Mukhopadhyay, M. Mitra, S. Mitra, Measurement **45**, 1651 (2012).
[7] S. K. Mukhopadhyay, M. Mitra, S. Mitra, Proc. of International Conference on Computer, Communication and Electrical Technology – ICCCET 2011, Tirunelveli, Tamilnadu, India, 143 (2011).

[8] C. T. Ku, H. S. Wang, K. C. Hung, Y. S. Hung, IEEE Transactions on Biomedical Engineering, **53**(12), 2577 (2006).
[9] N. Thakor, Y. Sun, H. Rix, P. IEICE Transaction on Information and Systems, E76-D, 1462 (1993).
[10] B. Bradie, IEEE Transactions on Biomedical Engineering, **43**, 493 (1996).
[11] S. G. Miaou, H. L. Yen, C. L. Lin, IEEE Transactions on Biomedical Engineering, **49**(7), 671 (2002).
[12] M. L. Hilton, IEEE Transactions on Biomedical Engineering, **44**(5), 394 (1997).
[13] N. Ahmed, P. J. Milne, S. G. Harris, IEEE Transactions on Biomedical Engineering, BME-**22**(6), 484 (1975).
[14] L. V. Batista, E. U. K. Melcher, L. C. Carvalho, Medical Engineering & Physics, **23**, 127 (2001).
[15] S. L. Chen, H. Y. Lee, C. A. Chen, H. Y. Huang, C. H. Luo, IEEE Systems Journal, **3**(4), 398 (2009).
[16] E. Chua, W. C. Fang, IEEE Trans. Consumer Electronics, **57**(1), 267 (2011).
[17] C. A. Chen, S. L. Chen, H. Y. Huang, C. H. Luo, sensors, **12**(12), 16211 (2012).
[18] S. L. Chen, J. G. Wang, Electronics Letters, **49**(2), 91 (2013).
[19] Guei-An Luo, Shih-Lun Chen, Ting-Lan Lin, IEEE International Conference on Information, Communications and Signal Processing, 4 (2013).
[20] R. Srinivasan, B. A. Engel, Applied Engineering in Agriculture 7.6, 779 (1991).
[21] Sergio A. Alvarez, Computer Science Department.

*Corresponding author: rajkumarresearch@gmail.com