A Low Complexity On-Chip ECG Data Compression Methodology Targeting Remote Health-Care Applications

Bastin Joseph, Amit Acharyya and P. Rajalakshmi

***Abstract*— In this paper, we propose a novel low complexity on-chip ECG data compression methodology targeting remote health-care applications. This is to the best of our knowledge the first attempt for on-chip reliable data compression. The proposed methodology has been implemented targeting Application Specific Integrated Circuit platform at 1MHz at V$_{dd}$ 1.62V for UMC 130nm technology library with 16 bits system word-length. Furthermore the proposed methodology results in a faithful reconstruction which has been validated using MIT-BIH PTB-DB as well as our institute's health repository IITH-DB. On an average about 90% compression is achieved with more than 83% R^2 statistics, 98% Cross Correlation and about 99% Regression between the original and the reconstructed data signifying the diagnostic accuracy. Subsequently the proposed methodology is capable of storing approximately 47 hrs of data in the same on-chip memory when compared to that of 5 hours of continuous data in the state of the art which would lead to enhanced diagnosis and prognosis in remote health-care.**

I. INTRODUCTION

Remote continuous personalized health monitoring involves capturing information from the patients using mobile devices and transmitting it to a centralized facility. The fact that these mobile devices are battery powered calls for the development of low complexity architectural design. Continuous transmission of vital data consumes significant amount of power. In order to minimize this power consumption, data can be sent to the centralized facility on a medical practitioner's request or at regular intervals of time. This requires the on-chip storage of ECG data so that it can be sent on clinician's request or on occurrence of abnormal events. The amount of data that can be stored on-chip is limited. This demands efficient on-chip compression of the ECG data which enables storage of the data for longer duration. But the advantage of on-chip local storage of the ECG data with respect to the continuous transmission would be negated if the compression architecture consumed more power than that of the transceiver module. Therefore the technical challenge here is to propose a low complexity on-chip ECG compression technique which would accomplish the following goals.

- The same memory which stores *X* amount of data in the state of the art method should be able store *nX* amount of data, where *n»1* which would help better diagnosis and prognosis (benefit from medical science perspective).
- The power consumption of the chip should be less than the state of the art, so that the battery backup lasts longer (benefit from technical perspective).

Compression of any data would mean reduction in the number of data points with respect to the original signal. In the current scenario, as the analysis of the ECG signals are vital for the health of the patient it is important to ensure that there is no compromise on the quality of the reconstructed signal. Loss-less and near loss-less compression methods can ensure the quality of the reconstructed signal. However, the loss-less compression techniques proposed in the recent years gives low compression ratios which forced the designers to think about lossy compression techniques for ECG data compression[2]. The existing lossy compression approaches([3],[4],[5],[6]) can be classified into: a) Transform based methods [7],[8],[9] b) parametric methods and c)Direct methods [11],[12],[13]. They cannot directly be targeted for on-chip architecture implementation or even implemented, lose their advantages because of no consideration of hardware implementation while designing the algorithm.This motivates us to introduce the concept of "token based" methodology (Section II) in order to achieve the low complexity on-chip compression, which on an average gives 90% compression. Subsequently its embedded architecture has also been implemented and its area and power analysis(Section III) are also presented.

II. PROPOSED TOKEN BASED METHOD AND ARCHITECTURE

A. Preprocessing

Daubechies, Haar and Legendre wavelets are frequently studied for compression purposes using Discrete Wavelet Transform(DWT)[15]. In this paper we have used Haar wavelet, because it can be easily implemented in hardware using simple adders and subtractors [14]. The studies made on the DWT coefficients of ECG data showed that most of the energy is contained in the lower frequency components at higher decomposition levels [15]. Selective thresholding is done on the DWT coefficients at different levels thus obtained, which are then compressed using the proposed algorithm(Section II-B). It has already been studied in [15] that the percentage of total information contained in the detail DWT coefficients are 0.00267%, 0.00925%, 0.002545%, 0.05621%, 0.09113% at level 1 to 5 respectively. Whereas the approximation coefficients at level 5 alone accounts for 99.81528% of total energy.Performing selective thresholding on the DWT coefficients results in a large number of zeroes among the coefficients. This motivates us to propose a

B.J.,A.A. and P.R. are with Dept. of Electrical Engineering, Indian Institute of Technology Hyderabad, 502205, India, with email {ee11b006,amit_acharyya,raji}@iith.ac.in respectively.

This work is partly supported by the DEITY, India under the "IOT for Smarter Healthcare" under Grant No: 13(7)2012-CC&BT, Dated 25th Feb, 2013.

method to store the signal in a more compact manner, instead of storing all the samples (including the large number of zeroes).

B. Proposed On-Chip Compression Methodology

The proposed compression method (Fig 1(a)) works on the DWT coefficients on which selective thresholding is applied. In this method both the non-zero data and its position (called as *"token"* in this paper) in the signal are stored and all the zeroes are ignored. Since the position of non-zero values are known, it is possible to reconstruct the signal by adding zeroes to the positions which have not been stored anywhere. In the current state of the art methodologies the zeros obtained after selective thresholding are not considered for algorithm implementation. However if implemented on-chip they will lose the advantages claimed in the algorithm. Each sample is compared with zero. The data is ignored if it is a zero. Data is stored if it is a non-zero. Along with the data its position is also stored. Both non-zero data and their position are stored in separate memory. The position of the data points is stored in a memory of width *ceil(log(n))*, so that any position in a frame of size *n* can be represented in this memory. Fig.2 (a) shows a general array of data and its compressed form is given in Fig.2 (b). Here '*d*'s refer to non zero data points and '*p*'s refer to the corresponding positions.

Number of bits consumed by the original signal is $w * n$ and compressed signal is $(w + ceil(\log n)) * nz$, where *w* is the word length for data points and *n* is the frame size and *nz* is number of non-zeros data points in DWT coefficients after selective thresholding.

For example, consider a set of data of 15 samples:

[0; 0; 0; 1.25;-2.25; 0; 0; 0; 0; 0; 0; 0;-1.23; 0.69; 1.72]

The compressed form of the corresponding samples will be:
Data: [1.25;-2.25;-1.23; 0.69; 1.72]
Position/Token: [4; 5; 13; 14; 15]

As shown above, it is to be noted that after compression, we have 5 non-zero values in the data memory and their corresponding positions in position memory. Assume word-length of data to be 16 bits. However, the length of the tokens will depend on the frame size. Here 4 (*ceil(log(15))*) bits are enough to indicate any token within set of 15 data samples. So we have 5 such 4-bit tokens and 5 16-bit data samples which amount to (*5 data * 16 bit + 5 token * 4 bit*) 100 bits in total. In the default case storing the data would have taken (*15 data * 16 bit*) 240 bits.

At the receiver's end, the compressed signal is reconstructed (as shown in Fig1 (b)) from the data in data memory and position(token) memory. Data is read from both the memories(data and position) simultaneously. A count of the number of samples reconstructed is maintained. If the count is less than the current position read, zeroes are added to the reconstructed signal. This is done till both the count and the position become equal. At this stage the current non-zero data from the data memory is added to the reconstructed signal. Once this is done the next data and position are read from memory and the process is repeated till all the data is

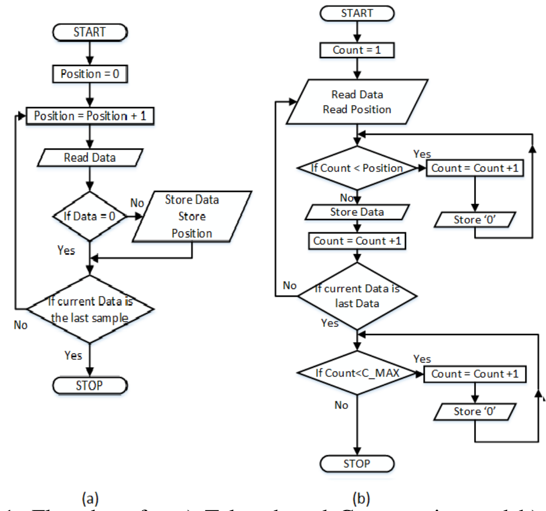

Fig. 1: Flowchart for a) Token based Compression and b) Token based Reconstruction

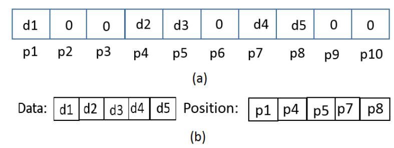

Fig. 2: a) Sample array for Token compression and b) Compressed array obtained after Token based compression

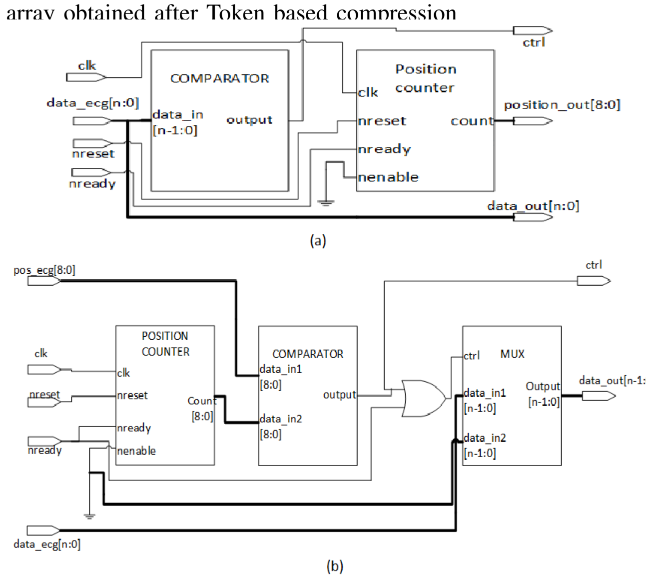

Fig. 3: Proposed Architecture for Token based (a)compression and (b)reconstruction

read from the memory. After adding all non-zero data points zeroes are added to the signal till the count is equal to the total number of samples in the original signal. This is done to incorporate the cases when the signal ends with a run of zeroes.

C. Proposed On-Chip Compression Architecture

The compression architecture (Fig 3 (a)) comprises of only a comparator to compare the data samples to zero and a counter to keep count of the position. The comparator compares the input ECG data with zero and produces *0* if data is non-zero and *1* if data is zero. Its output is given as the output *ctrl* signal. The block *position counter* keeps count of the position of the data. Both *data out* and *position out* are stored when *ctrl* is *0*.

At the receiver's end, the reconstruction architecture (Fig 3 (b)) contains a position counter which keeps the count of the signal being reconstructed. The comparator is used to compare the position of reconstructed ECG data with the output of the counter. It produces an output *1* if they are equal, else produce *0*. *nready* is a signal which will remain *1* till all the data has been written. It is *OR*ed with the output of the *comparator* to obtain the control signal for the mux. The mux gives *0* to *data out* channel if the *ctrl* signal is *0* else it gives the stored non-zero data. The latter happens when the count equals the position of the currently read data. At the end, once all the data has been read from the memory the output is made *0* and the counter counts till the count reaches *(framesize-1)*.

III. RESULTS AND DISCUSSION

A. Quality Parameters

a) Percentage Compression can be represented as:

$$\%Compression = \frac{N_{bc} - N_{ac}}{N_{bc}} * 100 \tag{1}$$

where N_{bc} and N_{ac} are the number of data points in the original ECG signal and compressed signal respectively.
b) Cross Correlation (CC):

$$CC = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})(\hat{x_i} - \bar{\hat{x}})}{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2}\sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{x_i} - \bar{\hat{x}})^2}} \tag{2}$$

where x and $\hat{x}$ are the reconstructed and the original ECG signal respectively and $\bar{x}$ and $\bar{\hat{x}}$ are their respective means.
c) R^2 Statistics :It is given as

$$R^2 coeff. = 100 \times \frac{(1 - (\|\boldsymbol{X} - \boldsymbol{Y}\|)}{(\|\boldsymbol{X} - \bar{\boldsymbol{X}}\|)} \tag{3}$$

where $\boldsymbol{X}$ and $\boldsymbol{Y}$ denote the original and reconstructed signal samples arranged as column vector and $\bar{\boldsymbol{X}})$ denotes the column vector formed by multiplying mean of original samples by a column vector having all entries as 1 and of length same as $\boldsymbol{X}$.
d) Regression : It can be defined as :

$$r_x = \left\{ \frac{\Sigma(Measured\ Sample\ ) \times (Derived\ Sample\ )}{(\Sigma(Measured\ Sample\ )^2 \times \Sigma(Derived\ Sample\ )^2)^{\frac{1}{2}}} \right\} \tag{4}$$

B. Experiment and Implementation Results

The Token based approach proposed in section II has been tested with real ECG data from PTB DataBase and IITH DataBase to prove its effectiveness. 25200 frames of data with 1024 [14] samples each belonging to 7 patients (with 15 leads each) has been taken from PTB DataBase for analysis. Apart from this ECG data of 30 patients obtained from IIT Hyderabad DataBase has been analyzed. DWT was performed on each frame of data(till 5 levels). Selective thresholding was performed on the DWT coefficients at different levels[15]. The threshold values for different levels were chosen as follows. For the detailed coefficients the threshold were 64%, 32%, 16% and 8% of the maximum value for level 2 to 5 respectively. Level 5 approximation coefficients are retained as they contain 99.81% of the total information in the signal.

TABLE I shows the maximum, mean and minimum % Compression achieved using the Token based approach for a word-length of 16. It gave a mean percentage compression of 89.52 for PTB-DB and 89.28 for IITH-DB.
Consider an ECG machine taking 1024 samples of 16 bit every second. Storing the ECG data of 5 hours would require (*2*1024*60*60*5*) *36864000* bytes or *35.15* MB memory. The same amount of memory (*35.15*) can be used to store the data for longer duration when in compressed form. As given in TABLE I on an average the proposed token based compression enables storing 47 hours of data in the same memory that would have been consumed in 5 hours if data were to be stored without compression. Another implication would be, if 5 hours data are expected to be stored in an on-chip memory, the proposed compression methodology requires only 3.83 MB memory when compared with the 36 MB memory needed by the state of the art methods.

Graph in Fig 5 (c) and (d) shows the mean % Compression for different word-length of the ECG data. It can be seen that % Compression increases as the word-length increases. This is because when the word length is increased, the position of the ECG samples are stored in a memory of finite length(*ceil(log n)*).
The error in the reconstructed signal is only due to the selective thresholding of DWT coefficients. But this has been minimized by the proper selection of threshold values. Based on the analysis on ECG data samples, a scaling factor of 10 is chosen to be optimal for a word-length of 16. This scaling factor minimizes the error due to fixed point implementation. TABLE I gives the performance matrices calculated for ECG data from PTB-DB and IITH-DB. As we can see, the Token based approach gives average Cross Correlation of 99.43 and 98.57 for PTBDB and IITHDB respectively. Similarly they give Regression of 0.9907 & 0.9868 and R^2 statistics of 90.41 & 83.76 for the two databases. Fig 4 (a) and (b) shows the original ECG signal and the signal reconstructed by token based approach for data from PTB-DB and IITH-DB respectively
The proposed algorithm after verifying in MATLAB was implemented in VHDL. VHDL programming was done in Modelsim. The architecture designed in VHDL was later synthesized in Cadence RTL Compiler. The analysis was done for 90nm for 1.62V at 125°C.
Fig 5 a) shows the area required for the compression and reconstruction block. As observed both the block occupy very less chip area. The area required for compression block increased only by 20.1% for an 8-fold increase in the word-length. Even area for the reconstruction block only doubles for an 8-fold increase increase in the word-length.
The calculation of power consumption by the designs is shown in Fig 5 b). Both the compression and reconstruction block consumes very less energy. The power consumption of the compression block remains almost constant even when the word-length is made 8 times. Similarly the power for reconstruction block increases only by 15% when the word

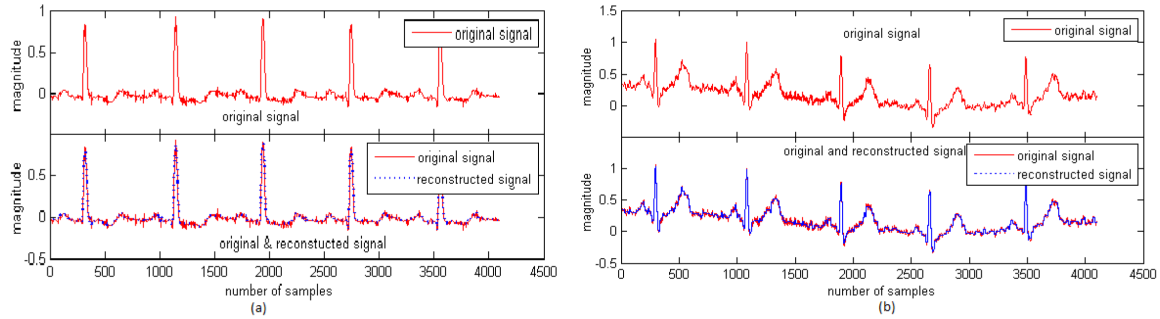

Fig. 4: Original and reconstructed signal after proposed methodology on a) PTB-DB and b) IITH-DB

PTB-DB									IITH-DB								
Time (hrs)			% Compression			Performance Metrics			Time (hrs)			% Compression			Performance Metrics		
$Mean$	Max	Min	$Mean$	Max	Min	CC	$Regression$	R^2	$Mean$	Max	Min	$Mean$	Max	Min	CC	$Regression$	R^2
47.71	76.22	18.73	89.52	93.44	73.30	99.43	0.9907	90.41	46.64	53.71	40.45	89.28	90.69	87.64	98.57	0.9868	83.76

TABLE I: Proposed compression leading to longer duration of data storage compared to state of the art's 5 hours data and compression performances observed for the proposed token based approach for a word length of 16 bits

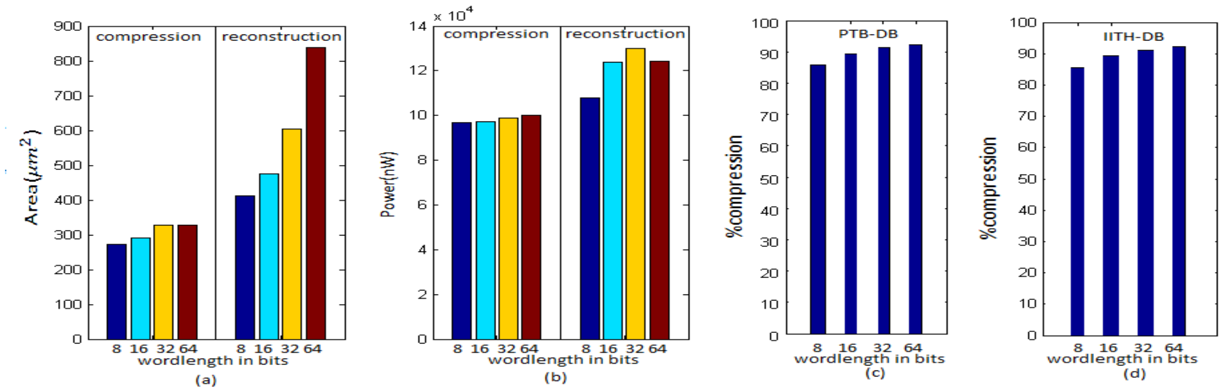

Fig. 5: a) Area, b) power reports for the proposed compression and reconstruction architecture for different word length. % Compression achieved using Token based approach for data samples from c)PTB Database and d)IITH Database

length changed from 8 to 64.

IV. CONCLUSION

In this paper we proposed an on-chip ECG compression methodology which have been shown to achieve an average % Compression of 90% and still maintaing the quality while reconstruction with CC and Regression around 98.57 and 0.9868 respectively for real ECG data from PTB-Database and IITH-Database. The proposed architecture also promises low complexity and low power consumption on-chip implementation which are vital for battery powered mobile devices.

REFERENCES

[1] World Health Organization
[2] Jalaleddine, Sateh MS, et al. ECG data compression techniques-a unified approach. Biomedical Engineering, IEEE Transactions on 37.4 (1990): 329-343.
[3] Acharyya, Amit, et al. 'Memory reduction methodology for distributedarithmetic-based DWT/IDWT exploiting data symmetry. Circuits and Systems II: Express Briefs, IEEE Transactions on 56.4 (2009): 285-289.
[4] M. Abo Zahhad and B. Rajoub, "ECG cmpression algorithm based on coding and energy compaction of the wavelet coefficients," in IEEE ICECS, vol. 1, 2001, pp 441 -444 vol.1.
[5] A. Ebrahimzadeh and M. Azarbad, 'Ecg compression using Wavelet transform and three-level quantization,' in IDC, Aug 2010, pp 250-254.
[6] S. Jalaleddine, C. Hutchens, R. Strattan, and W. Coberly, 'Ecg data compression tecniques-a unified approach,' IEEE Trans. Biomed. Eng.,vol. 37, No. 4, pp.329 -343, Apr 1990.
[7] M. E. Womble and A. M. Zied, 'A statistical approach to ECG/VCG data compression,' in Optimization of Computer ECG Processing Halifax, N.S., Canada, 1980, pp. 91-101.
[8] S. Olmos, M. Millan, J. Garcia, and P. Laguna, 'ECG data compressionwith the Karhunen-Loeve transform,' in Proc. Comput. Cardiol., Indianapolis, , IN, Sept. 1996, pp. 253-256
[9] B. R. S. Reddy and I. S. N. Murthy, 'ECG data compression using Fourier descriptions,' IEEE Trans. Biomed. Eng., vol. 33, pp. 428-434, Apr. 1986.
[10] P. S. Hamilton, 'Adaptive compression of the ambulatory electrocardiogram,' Biomed. Inst. Technol., vol. 27, no. 1, pp. 56-63, Jan. 1993.
[11] Nygaard, Ranveig, Gerry Melnikov, and Aggelos K. Katsaggelos. "A rate distortion optimal ECG coding algorithm." Biomedical Engineering, IEEE Transactions on 48.1 (2001): 28-40.
[12] D. A. Dipersio and R. C. Barr, 'Evaluation of the fan method of adaptive sampling on human electrocardiograms,' Med. Biol. Eng. Comput., pp. 401-410, Sept. 1985.
[13] J. Abenstein and W. Tompkins, 'New data-reduction algorithm for real-time ECG analysis,' IEEE Trans. Biomed. Eng., vol. BME-29, pp. 43-48, Jan. 1982.
[14] Mazomenos, E., Biswas, D., Acharyya, A., Chen. T., Maharatna, K., Rosengarten, J., Morgan, J and Curzen, N. (2013), A Low-Complexity ECG Feature Extraction Algorithm for Mobile Healthcare Applications, IEEE Journal of Biomedical and Health Informatics. , Vol. 17, No. 2, March 2013
[15] Biswas, Dwaipayan, Evangelos B. Mazomenos, and Koushik Maharatna. ECG compression for remote healthcare systems using selective thresholding based on energy compaction. Signals, Systems, and Electronics (ISSSE), 2012 International Symposium on. IEEE, 2012.