VLSI Implementation of ECG Compression Algorithm using Golomb Rice Coding

Tsung-Han Tsai, Muhammad Awais Hussain, Ping-Zen Hao
Department of Electrical Engineering, National Central University, Taiwan

***Abstract*—A lossless electrocardiogram compression algorithm has been implemented in VLSI circuit design. This structure is based on memory-less model i.e. no RAM is being used in the design. Adaptive linear prediction has been combined with Golomb Rice coding to achieve the lossless data compression. Overall compression ratio for MIT-BIH Arrhythmia database is 3. The design has been synthesized using TSMC 0.18*μm* technology. The implementation results show high working speed with low power while working at the processing speed of 90 MHz.**

I. INTRODUCTION

Data compression has been one of the important part in modern world applications. It is required in different fields which vary from communication to medical to data storage in giant industries. In medical field, data communication is being used in many departments such as pathology, radiology and cardiology etc.

Electrocardiography (ECG) is an important process which is used in medical field. It describes the electrical activity of the heart over a specific period of time. To find some diseases, lot of ECG data is required to evaluate. As a result, lot of data is required to store as ECG recoding duration may vary from hours to days. So data compression is required to store huge ECG data.

To compress the ECG data, many lossy and lossless techniques have been developed. A review of many lossless and lossy techniques can be found in [1] and [2]. ECG data is quite sensitive as it will help to identify the heart activity, so lossy data storage may have effect on original data as some data will be removed during compression process. In this study, VLSI implementation of lossless ECG compression using Golomb Rice coding has been performed.

II. ARCHITECTURE

For lossless compression, Golomb Rice coding has been combined with Adaptive Linear Prediction (ALP). ALP is playing a role of predictor, which is an essential part of compression. After ALP, error prediction is being performed. ALP output is processed through Golomb Rice coding module and then final packing is performed in packing module.

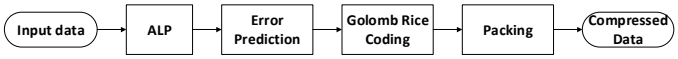

Fig. 1: Block Diagram of proposed ECG compression algorithm

Figure 1 shows the block level flow of encoding process for ECG compression algorithm. For decoding process, same blocks are applied in reverse order i.e. unpacking, Golomb Rice coding, error prediction, ALP and then original data is achieved.

Original input data is 11 bits which go through ALP module. ALP module provides the output of predicted values and a signal for Golomb Rice coding module, which is a flag that Golomb Rice module should start working on the output data of ALP as it is valid data. In the same time, error calculation is performed on the output of ALP module. This module has been implemented in combinational technique so it performs instantaneous processing as soon as output comes from ALP module.

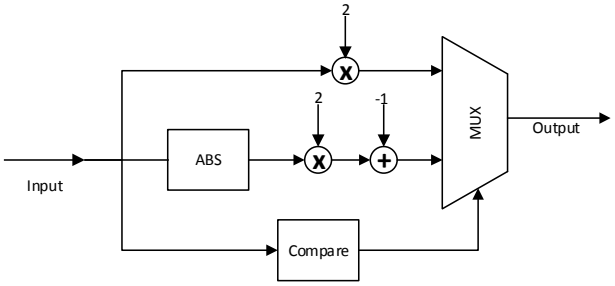

Fig. 2: Proposed architecture for error calculation

For Golomb Rice coding, memory-less architecture has been designed. Output of Golomb Rice module depends upon incoming input. As window size of 40 has been utilized for incoming data so it will process data after 40 inputs have arrived. Fig. 3 shows architecture of Golomb Rice coding.

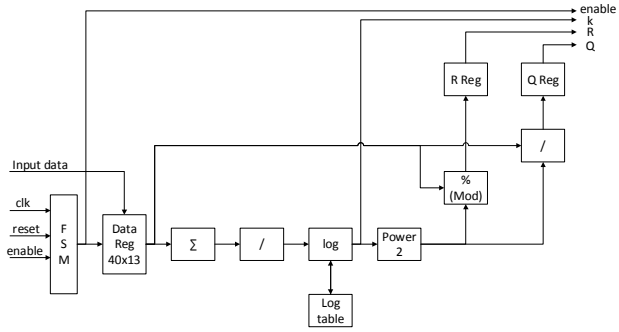

Fig. 3: Proposed architecture for Golomb Rice Code

After Golomb Rice coding, packing is being performed. Packing module has been implemented using finite state machine. A number of inputs are coming to this finite state machine which have been shown in fig. 4. *Q* and *R* values, which are coming from Golomb Rice module, are being packed together. 0 is being inserted between *Q* and *R* value to differentiate between value of *Q* and *R*. Packing structure will look like $Q_0 0 R_0$, $Q_1 0 R_1$, ... for different values of *Q* and *R*, after packing has been performed. The values of *Q* and *R* differ for different inputs so these are being stored in a temporary register. Whenever this *'specific'* register gets 16-bits, these 16-bits are sent as output. While *valid* signal is a flag which

informs that output is valid or not and *finish* flag becomes one once all data has been processed.

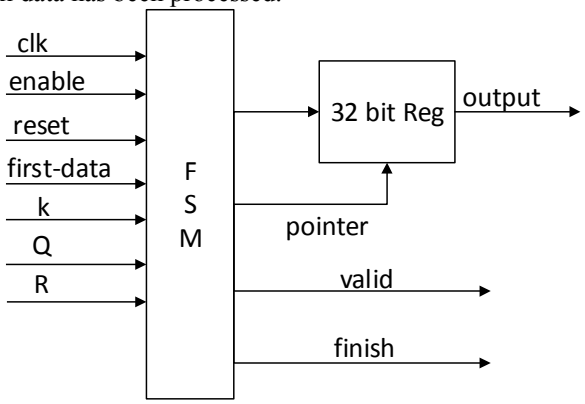

Fig. 4: Packing module architecture

III. RESULTS

VLSI design has been implemented using TSMC 0.18 μm technology. The designed is working at frequency of 90 MHz due to which power consumption is high. If the design is run at 10 MHz, then power consumption drops to 1.15 mW. Moreover, this design is using large power due to memory-less design as no RAM has been utilized to store intermediate data at different stages. All values are being processed through registers. For testing purposes, MIT-BIH database has been used. For MIT-BIH database, compression ratio of the proposed design is 3.

TABLE I. Design Specifications

Parameter	Proposed Work
Compression Ratio	3
Technology	0.18 µm TSMC
Core Voltage	1.8V
Working frequency	90 MHz
Core area	1.2 x 1.2 mm^2
Gate count	22K
Number of I/O pins	30
Power Consumption	6.7 mW

Table II. shows comparison of proposed work with some implemented ECG compression techniques. This work's design frequency is higher than [3] and [4], which provides the benefit of executing design in real time environment at high frequency. Although [4] has less core area as compared to proposed design but working frequency of proposed design is better than [4]. Power consumption is less in other design but it comes with the tradeoff of frequency as implemented technique has considerably high frequency as compared to other techniques in table II.

Table II. Comparison with related work

Parameter	Proposed Work	Ieong et. al. [3]	Chen et. al. [4]
Technology	0.18 µm TSMC	0.18 µm CMOS	0.18 µm
Core Voltage	1.8V	0.45 V	1.8 V
Working frequency	90 MHz	1 KHz	50 MHz
Core area	1.2 x 1.2 mm^2	NA	0.85 x 0.85 mm^2
Gate count	22K	19.5K	NA
Power Consumption	6.7 mW	375 nW	2.3 mW

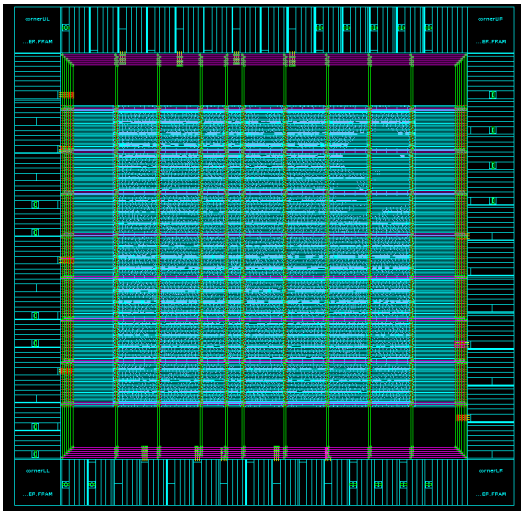

Fig. 5: Layout of lossless ECG compression algorithm

IV. CONCLUSION

In this study, a lossless ECG compression scheme has been implemented using Golomb Rice coding and ALP. The design does not use any RAM to store data. VLSI implementation has been performed using TSMC 0.18μm technology. When compared with other implementations, it shows overall increase in performance parameters.

REFERENCE

[1] A. Malik, "Compression Techniques for ECG Signal : A Review," *Int. J. Mod. Electron. Commun. Eng.*, no. 4, pp. 1–4, 2016.

[2] B. Singh, "A Review of ECG Data Compression Techniques," *Int. J. Comput. Appl. (0975 – 8887)*, vol. 116, no. 11, pp. 39–44, 2015.

[3] C.-I. Ieong, M. Li, M.-K. Law, P.-I. Mak, M. I. Vai, and R. P. Martins, "A 0.45 V 147–375 nW ECG Compression Processor With Wavelet Shrinkage and Adaptive Temporal Decimation Architectures," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 25, no. 4, pp. 1307–1319, Apr. 2017.

[4] S.-L. Chen *et al.*, "A Power-Efficient Mixed-Signal Smart ADC Design With Adaptive Resolution and Variable Sampling Rate for Low-Power Applications," *IEEE Sens. J.*, vol. 17, no. 11, pp. 3461–3469, Jun. 2017.