IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁙ Peer-reviewed & Refereed journal ⁙ Vol. 13, Issue 5, May 2024

DOI: 10.17148/IJARCCE.2024.13576

“Lossless Compression and Implementation of medical signals using verilog”

Madhukara S[1], Bhavana H N[2], Arshitha M[3], Chinnadevara Kushal K[4]

Assistant Professor, Dept of Electronics and Communication, SJC Institute of Technology, Chickballapur, Karnataka, India[1]

Dept of Electronics and Communication, SJC Institute of Technology, Chickballapur, Karnataka, India[2-4]

Abstract: The sending/receiving of data (data communication) is the most power consuming in wireless communication, wireless sensor networks (WSN), bio medical devices and data storage since the electronic components at transmitter end are depending on batteries not generally rechargeable characterized by limited capacity. Data compression is among the techniques that can help to reduce the amount of the exchanged data between wireless sensor nodes in bio medical devices resulting in power saving. Nevertheless, there is a lack of effective methods to improve the efficiency of data compression algorithms and to increase transmission reception energy efficiency. In this paper, we proposed a novel lossless compression approach for ECG data compression using Transition Inversion based Run Length Encoding algorithms. TIE -RLE is an optimization of the RLE algorithm, which aims to improve the compression ratio. This method will lead to less storage cost and less bandwidth to transmit the data, which positively affects the sensor nodes’ lifetime and the network lifetime in general. The proposed scheme increases run length of number zeroes and reduces reduced number of one's transmission which reduces power consumption and increases compression ratio of ECG transmission and storage. The proposed architecture is implemented using verilog HDL and simulation/synthesize was done in Modelsim and Xylinx vivado tools.

Keywords: renewable and non-renewable, walking or jogging, generate power, piezoelectric sensor, noiseless and pollution-free.

I. INTRODUCTION

System reconfiguration is a valuable technique in embedded technology, Networking, Communication and Image Processing. Field Programmable Gate Array offers a broad platform for hardware realization of application-specific algorithms. In SoC, Network Processors, Image Processing Units or any Digital System Design, the Program code is developed using High-Level Languages and stored in Program memory or External memory. This will be fetched out and placed in the FPGA board during execution, along with the required details, is known as Configuration.

The frequent change of modules is allowed in system design for further modifications and improvement. This is called reconfiguration. The reconfiguration data is also retrieved and stored in memory. Hence, there is a huge demand for memory in FPGA design and reconfiguration. To improve the memory size and to reduce the memory requirements, the stored information is compressed using preferable techniques. These techniques improve communication bandwidth and thereby decrease the reconfiguration time. In the FPGA system design, the data is expected to be retrieved without any loss.

Bitstream Configuration

In general, an external entity downloads the configuration on FPGA via one of the configuration interfaces. Usually, FPGAs are brought-up with Read-Only Memory (ROM), where configuration data is normally stored. When power-on, this data is downloaded into FPGA configuration memory. This configuration data is called bitstream. This can either be fully or partially downloaded based on the extent of the configuration memory.

Bitstream is a binary file in which configuration information for a particular device is stored. The data has to be copied onto the Field Programmable Gate Arrays - Static Random Access Memory (FPGA - SRAM) to configure the device’s functionality. A full bitstream configures the whole configuration memory of the device, and it is used for static configuration at the power-up of FPGA. Partial bit-streams are generated through partial reconfiguration design flow and, it configures only a portion of the device. Irrespective of being the full or partial configuration of the bitstream, it has a fixed pattern.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2024

DOI: 10.17148/IJARCCE.2024.13576

Data compression

Data Compression is also known as bit-rate reduction. Data compression permits sending a data object or file quickly, over a network or the Internet, optimizing physical storage resources. The compression techniques are used to reduce the memory requirements for storing configuration data. Data compression works through several compression techniques and software solutions that utilize Data compression algorithms to reduce the data size. It is the process of modifying the data that is encoding or converting the bit structure of data in such a way that it consumes less space in memory. The redundant data is identified and compressed using a specific method that reduces the memory space. The compression techniques are used to reduce the memory requirements for storing the configuration bitstream. All the techniques of data, compression identify and minimize or remove the amount of data redundancies present. For a given application, a subset of operations may get repeated in successive configurations consistently. These translate into configuration data that gets repeated in multiple bit-streams. The reconfiguration process of FPGA provides opportunities to exploit these redundancies through data reuse in successive configuration bit-streams.

II. PROPOSED SYSTEM ARCHITECTURE

The existing work is a combination of Dictionary and Bit-mask method with Modified Run-Length-Rice Coding (MGC). In this proposed work, for further improvement, the combined sequence of compressed bit-streams using Dictionary with Bitmask is compressed using Run-Length coding withreduced bit codes known as Advanced Run-Length Coding (ARLC) and the resultsare verified using Vertex-IV family FPGA boards. (Virtex-4 FPGA UserGuide: Virtex-II Pro User Guide).

The proposed Advanced Run-Length Coding (ARLC) makes use of the existing Run-Length for better enhancement of the compression efficiency. The first step in is to create a table with the runs of zeros until the code that ends with one is created. The successive 0‟s are moderate in size otherwise creating a table itself is a complicated process. The parameter, m, decides variable-length code structure of data in a table which has a direct impact on the compression efficiency also. The parameter $m = 2^k$, m is always the digit with the power of 2. This produces the code with equal length. The GR bit positioning is explained with the sample data $m=2^2=4$ in order to reduce the complexity. If $m=2^1=2$ the GR codes are minimal where m=4 gives 16 combinations.

The representation of run length for subsets is shown as in Figure.

Data subset	*001*	*00001*	*01*	*0000001*	*0001*
Run – length	2	4	1	6	3

Figure 3.1 Run-length with Data set

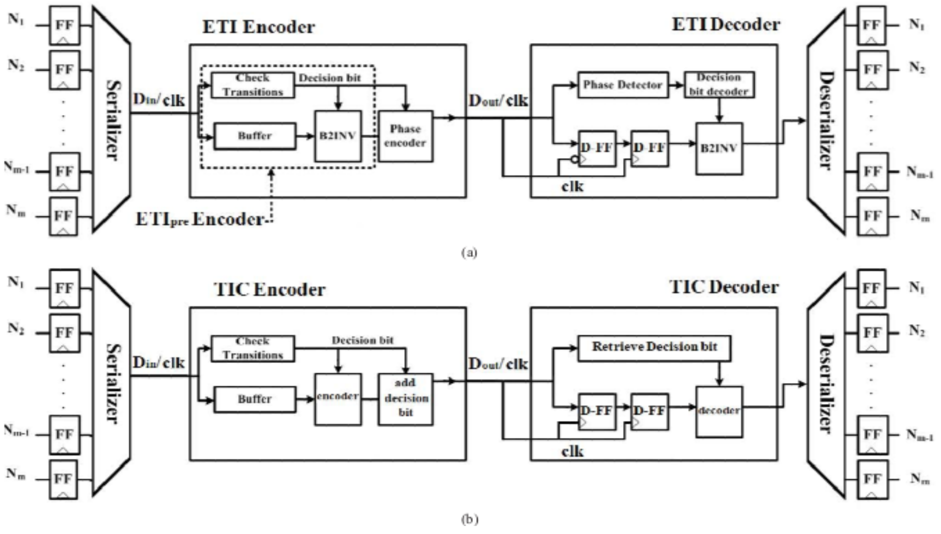


Fig. 1: a. Encoder b. Decoder

DOI: 10.17148/IJARCCE.2024.13576

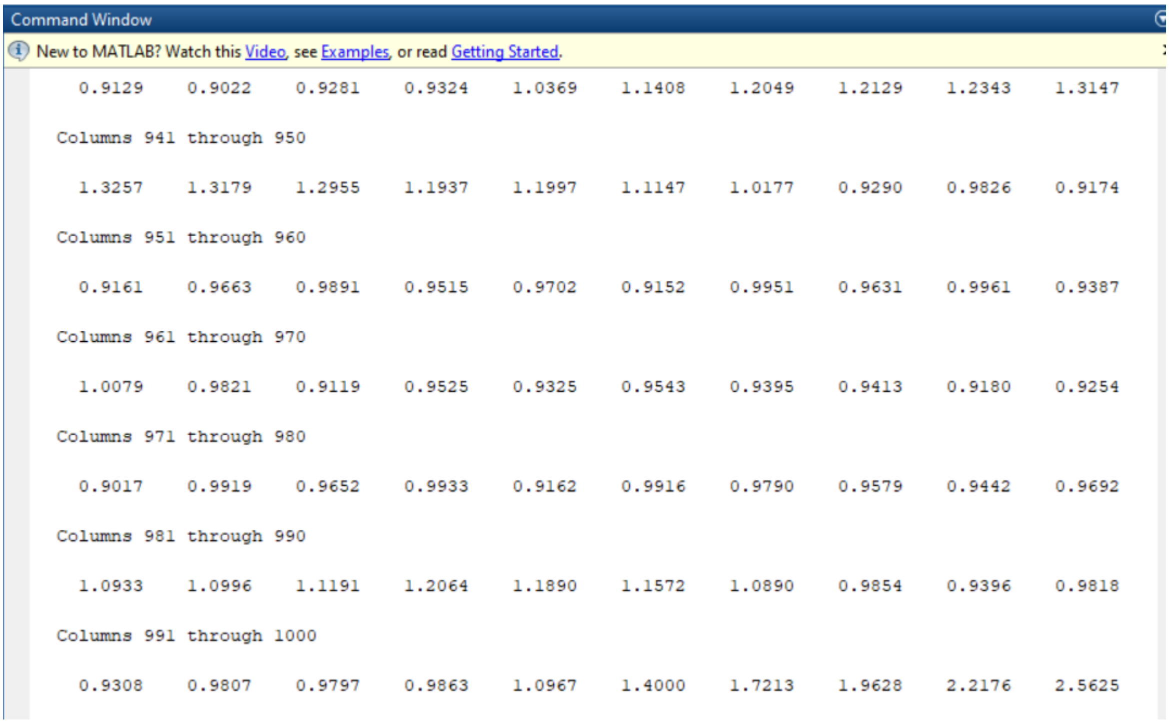
The overall architecture of the ETI scheme is shown in Fig(a). We add the ETI pre-encoder block and the TIC architecture for clarity. The ETI pre encoder does not provide the decision bit information so it cannot be decoded in the receiver. The ETI pre encoder is shown by the dashed box in the ETI encoder in figure.

The TIC counts the transitions in the data word then uses this information to perform encoding. The transition indication bit is added to every data word to indicate whether there is an inversion or not. The decoder adopts the transition indication bit to perform the decoding as shown in Fig(b). In the ETI encoder part, the input data Din are stored in the buffer to wait until the check transition operation is completed. The transition and threshold in a data word are used to set the decision bit.

III. SOFTWARE REQUIREMENTS

1 *Matlab*:

Matlab is used to generate an ECG signal with a dynamic heart rate and adds baseline wander and noise to mimic a real-world ECG signal. The ECG signal is converted into binary samples in a readable binary text file. R-pea and QRS peaks are generated, and the heart rate (BPM) is calculated and displayed, along with ECG signal plots. This script can used to generate ECG signal processing and analyze heart rate variations over time.



New to MATLAB? Watch this Video, see Examples, or read Getting Started.

```
    0.9129    0.9022    0.9281    0.9324    1.0369    1.1408    1.2049    1.2129    1.2343    1.3147

  Columns 941 through 950

    1.3257    1.3179    1.2955    1.1937    1.1997    1.1147    1.0177    0.9290    0.9826    0.9174

  Columns 951 through 960

    0.9161    0.9663    0.9891    0.9515    0.9702    0.9152    0.9951    0.9631    0.9961    0.9387

  Columns 961 through 970

    1.0079    0.9821    0.9119    0.9525    0.9325    0.9543    0.9395    0.9413    0.9180    0.9254

  Columns 971 through 980

    0.9017    0.9919    0.9652    0.9933    0.9162    0.9916    0.9790    0.9579    0.9442    0.9692

  Columns 981 through 990

    1.0933    1.0996    1.1191    1.2064    1.1890    1.1572    1.0890    0.9854    0.9396    0.9818

  Columns 991 through 1000

    0.9308    0.9807    0.9797    0.9863    1.0967    1.4000    1.7213    1.9628    2.2176    2.5625
```

Fig 2. Matlab ECG samples

2 *Modelsim:*

The $fopen function opens a ECG samplefile and returns a multi-channel descriptor in the format of an unsized integer. This is unique for each file. All communications between the simulator and the ECG samples take place through the file descriptor. Users can specify only the name of a file as an argument, this will create a file in the default folder or a folder given in the full path description.

To close an opened file use the $fclose function. This function is called without any arguments, it simply closes all opened files.All file output tasks work in the same way as their corresponding display tasks. (see the Display Tasks chapter for further information) The only difference is a file descriptor that appears as the first argument in the function argument list. These functions only can append data to a file and cannot read data from files.

DOI: 10.17148/IJARCCE.2024.13576

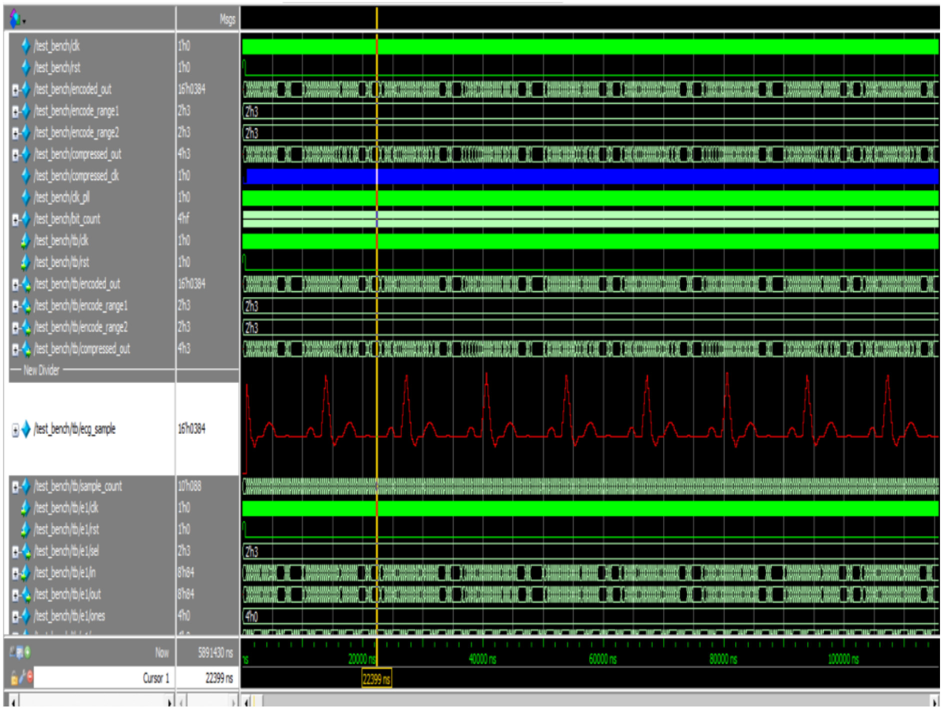

Msgs
/test_bench/clk 1'h0
/test_bench/rst 1'h0
/test_bench/encoded_out 16'h0384
/test_bench/encode_range1 2'h3
/test_bench/encode_range2 2'h3
/test_bench/compressed_out 4'h3
/test_bench/compressed_clk 1'h0
/test_bench/clk_pll 1'h0
/test_bench/bit_count 4'hf
/test_bench/tb/clk 1'h0
/test_bench/tb/rst 1'h0
/test_bench/tb/encoded_out 16'h0384
/test_bench/tb/encode_range1 2'h3
/test_bench/tb/encode_range2 2'h3
/test_bench/tb/compressed_out 4'h3
New Divider
/test_bench/tb/ecg_sample 16'h0384
/test_bench/tb/sample_count 10'h088
/test_bench/tb/e1/clk 1'h0
/test_bench/tb/e1/rst 1'h0
/test_bench/tb/e1/sel 2'h3
/test_bench/tb/e1/in 8'h84
/test_bench/tb/e1/out 8'h84
/test_bench/tb/e1/ones 4'h0
Now 5891430 ns
Cursor 1 22399 ns
20000 ns
40000 ns
60000 ns
80000 ns
100000 ns
22399 ns

Figure 3. ECG Sampling and Compression

3 *Xylinx Vivado*

In Xilinx vivado, the RTL View refers to the Register Transfer Level view, which provides a detailed representation of the design at the register transfer level. This view is useful for understanding the logic and structure of the design.

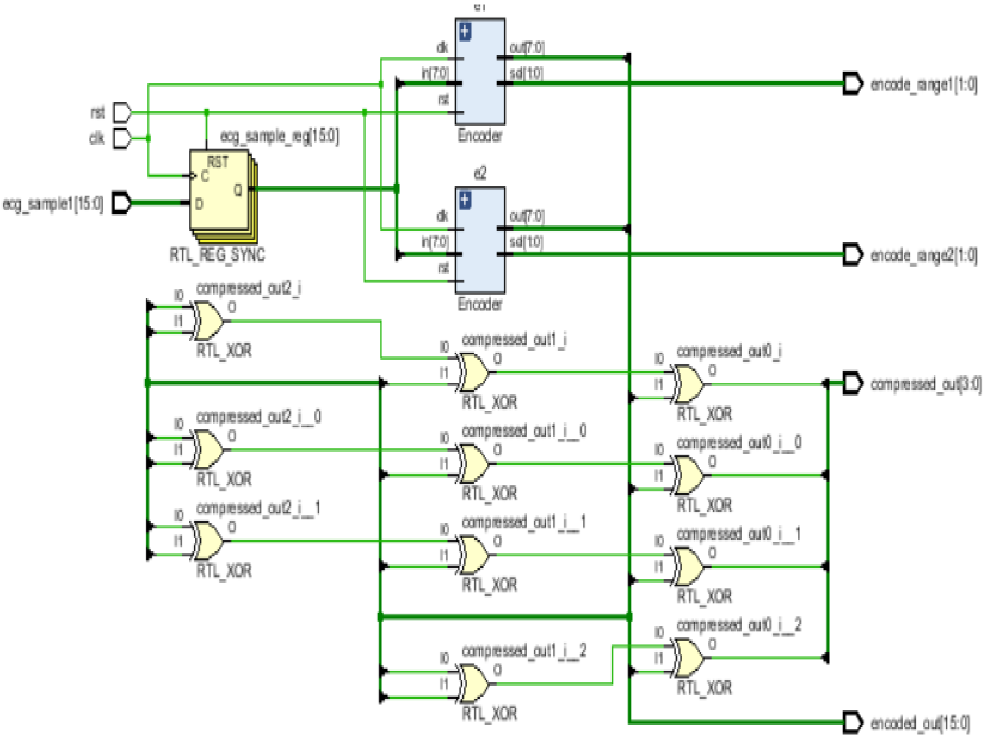


Fig 4. RTL view

DOI: 10.17148/IJARCCE.2024.13576

IV. CONCLUSION

Energy consumption was still the utmost important constraint in bio medical signal processing due to continuous monitoring of patients. It is difficult for the researchers to find new methods to reduce or at least rationalize energy consumption in bio medical devices. Data compression showed efficacy in reducing the data size to be transmitted, therefore consuming less energy to transfer the same data by the communication unit. In this paper, we proposed an optimized lossless compression approach, which based on embedded transition inversion encoding and a proposed TIE - RLE algorithm. Our approach showed a better improvement in compression up-to 60% compared to our previous work. On the other side, it performed good results comparing with the other traditional algorithms. In the future, we are going to implement ECG coefficient memory model to minimize ECG sample storage and compression ratio.

REFERENCES

[1] Sumathi Sokkanarayanan, Chithrakkannan. R., K. R. H. I. R. P. . (2024). Improving Medical Imaging Diagnostics by Utilizing Compression Techniques and Implementing FPGA Acceleration with Vivado HLS. International Journal of Intelligent Systems and Applications in Engineering

[2] Turcza, P., Duplaga, M. Low-Power Low-Area Near-Lossless Image Compressor for Wireless Capsule Endoscopy. Circuits Syst Signal Process 42, 683–704 (2023).

[3] Dang, F. (2023). Image lossless compression algorithm optimization and FPGA implementation. Frontiers in Computing and Intelligent Systems

[4] Athira M S , Rachana M K, 2022, Lossless EEG Compression based on Highly Efficient Multilevel Compression Method for VLSI Implementation, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 11, Issue 05 (May 2022).

[5] Kadam, Deepak. (2022). Verilog HDL Implementation of Efficient Lossless EEG Compression Architecture Based on VLSI. 2022.

[6] M. C. H. Zerva, V. Christou, N. Giannakeas, A. T. Tzallas and L. P. Kondi, "An Improved Medical Image Compression Method Based on Wavelet Difference Reduction," in IEEE Access, vol. 11, pp. 18026-18037, 2023.

[7] Banerjee, Soumyendu & Singh, Girish Kumar. (2022). A new real-time lossless data compression algorithm for ECG and PPG signals. Biomedical Signal Processing and Control.

[8] P. Luanloet, W. Suwansantisuk, and P. . Kumhom, "An Efficient Electrocardiography Data Compression", ECTI-CIT Transactions , vol. 17, no. 3, pp. 387–397, Sep. 2023.