SOME STUDIES ON THE IMPLEMENTATION OF WAVELET BASED ECG COMPRESSION TECHNIQUES INVOLVING BIT FIELD PRESERVING AND RUNNING LENGTH ENCODING ON TMS320C6713 DIGITAL SIGNAL PROCESSOR

A THESIS SUBMITTED

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE AWARD OF THE DEGREE OF

MASTER OF

ELECTRONICS & TELECOMMUNICATION ENGINEERING

of

JADAVPUR UNIVERSITY

by

RAHUL CHANDWANI

Roll No: M4ETC1602

Reg. No:128915 of 2014-15

Under the guidance of

PROF. SALIL KUMAR SANYAL

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGINEERING

JADAVPUR UNIVERSITY, KOLKATA-700032

INDIA

MAY, 2016

FACULTY OF ENGINEERING AND TECHNOLOGY

JADAVPUR UNIVERSITY

CERTIFICATE

This is to certify that the dissertation "**Some studies on the Implementation of Wavelet based ECG compression techniques involving Bit Field Preserving and Running Length Encoding on TMS320C6713 Digital Signal Processor**" has been carried out by **RAHUL CHANDWANI (University Registration No:128915 of 2014-15)** under my guidance and supervision and be accepted in partial fulfillment of the requirements for the degree of Master of Electronics and Telecommunication Engineering of Jadavpur University. The research results presented in the thesis have not been submitted to any other University/Institute for the award of any Degree or Diploma.

(**Prof. Salil Kumar Sanyal**)
Supervisor
Department of Electronics &
Telecommunication Engineering
Jadavpur University, Kolkata- 700032

COUNTERSIGNED

(Prof. P. Venkateswaran)
Head, Department of Electronics &
Telecommunication Engineering,
Jadavpur University,
Kolkata – 700 032

(Prof. Sivaji Bandyopadhyay)
Dean, Faculty of
Engineering & Technology,
Jadavpur University,
Kolkata – 700 032

FACULTY OF ENGINEERING AND TECHNOLOGY

JADAVPUR UNIVERSITY

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGINEERING

CERTIFICATE OF APPROVAL*

The foregoing thesis is hereby approved as a creditable study of an Engineering subject, carried out and presented in a manner satisfactory to warrant its acceptance as a prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein but approve the thesis only for the purpose for which it has been submitted.

Final examination and evaluation
of the Thesis of
Sri Rahul Chandwani

Signature of the External Examiner

Signature of the Supervisor

*Only in case the thesis is approved

FACULTY OF ENGINEERING AND TECHNOLOGY

JADAVPUR UNIVERSITY

DECLARATION OF ORIGINALITY AND COMPLIANCE OF ACADEMIC THESIS

I hereby declare that this thesis titled "**Some studies on the Implementation of Wavelet based ECG compression techniques involving Bit Field Preserving and Running Length Encoding on TMS320C6713 Digital Signal Processor"** contains literature survey and original research work by the undersigned candidate, as a part of his degree of **Master of Electronics and Telecommunication Engineering of Jadavpur University.**

All information have been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by these rules and conduct, I have fully cited and referenced all the materials and results that are not original to this work.

Name : RAHUL CHANDWANI

Examination Roll No : M4ETC1602

Thesis Title **:** SOME STUDIES ON THE IMPLEMENTATION OF WAVELET BASED ECG COMPRESSION TECHNIQUES INVOLVING BIT FIELD PRESERVING ON TMS320C6713 DIGITAL SIGNAL PROCESSOR.

Signature :

Date :

Acknowledgement

There are several people that I would like to thank for the support and encouragement throughout the year, without them this work would not have been possible.

First of all I would like to deeply express my sincere gratitude to my thesis supervisor, Professor Salil Kumar Sanyal, for his advice and suggestions that have guided me throughout the year. I feel grateful for having the opportunity to work and learn under him during the classes and also during my thesis work.

I would also like to thank Prof. P. Venkateswaran, Head, Department of ETCE and also former head Prof. Iti Saha Misra for their continuous encouragement in keeping myself motivated throughout the process of this thesis work. His help in the matter of timely procurement of laboratory facilities has immensely been valuable for the completion of this work.

I would also like to thank my laboratory co-workers for their critical analysis and support that has helped me in my work.

Last but not the least I would like to thank my parents and friends for their love and support without which I would not have been able to achieve anything in life.

Place: Kolkata

Date:

RAHUL CHANDWANI

Department of Electronics and
Telecommunication Engineering,
Jadavpur University, Kolkata-700 032

ABSTRACT

ECG signal is a well- stabilized diagnostic tool for cardiac diseases. Compression of ECG signal is important for Digital Holters recording, signal archiving, Telemedicine and for transmission over communication channels. Efficient ECG compression can reduce the payload of real-time ECG transmission and amount of data storage in long-term ECG recording. The present work introduces an effective real-time compression scheme to reduce the payload size of the transmission channel. This scheme utilizes the Discrete Wavelet Transform (DWT), Bit-Field Preserving (BFP) and Running Length Encoding (RLE) to get the efficient compression results. An attempt is made to implement this compression algorithm on Texas Instrument's TMS320C6713 Digital Signal Processor. The ECG signal compression based on DWT-BFP-RLE is implemented on this board, also ECG signal reconstruction from the compressed signal has also been implemented. This work mainly focuses upon how to get good ECG compression performance using above method using the DSP board, also the reconstruction error after the decompression should be as low as possible. The results have been obtained for different ECG records taken from MIT-BIH Arrhythmia database and their Compression Performance evaluation based on the different parameters have been done. Comparisons of this algorithm with other standard compression algorithms has also been done.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1
Introduction

1.1 Overview

An ECG signal is a graphical representation produced by an electrocardiograph, which records the electrical activity of the heart over time [1]. In recent years, ECG signal plays an important role in the primary diagnosis, prognosis and survival analysis of heart diseases. The ECG signal contains an important amount of information that can be exploited in different manners. It allows for the analysis of anatomic and physiologic aspects of the whole cardiac muscle.

Transmission of this ECG signal through communication channels are very important issue in many applications related to the clinical practice [2]. The volume of ECG data produced by monitoring system can be quite large over a long period of time and ECG data compression is often needed for efficient storage of the ECG data, so when we need to transmit ECG data for telemedicine applications, data compression needs to be utilized for efficient transmission. ECG signals are not only used in the hospitals but also they are used by paramedics responding to the accidents scenes in emergency vehicles. They are also used by the clinicians at remote sites. Wireless ECG transmission is necessary for physiological movement of the patients which are free moving and athletes or soldiers during training [3]. With efficient ECG compression, the amount of the data storage and payload of the real-time transmission can be reduced. The main goal of the compression is to reduce the number of digitized ECG signal without the loss of the diagnostic data. The compressed signal should not represent the exact reconstruction of the original ECG signal, in it only the important features of the ECG signals are preserved. The main important factors of the ECG signal compression are: (1.) The ability of reconstructing the important features from the compressed data. (2.) The compression ratio, (3.) Execution time. (4.) The amount of error between the original and the re-constructed ECG signal [4].

Discrete Wavelet Transform [5,6] is a very powerful time frequency-analysis tool that can be utilized for the compression of ECG signals. It can also be used for de-noising and feature extraction of the ECG signal. The DWT transforms the signal into time-frequency sub-bands, this make it easy to decompose the signal into different waveforms[7]. DWT also has the properties of multi-resolution and high-energy compaction [8]. This wavelet based technique suit well with the standard signal processing methods and encoding schemes to produce good compression results.

This DWT transforms the signal into sub-bands, that can be encoded using Compression techniques like Set Partitioning In Hierarchal Tree coding (SPIHT) [11] , Vector Quantization (VQ) [15], Energy Packing Efficiency (EPE) and other encoding schemes. Out of these compression schemes some are very complex to implement on FPGAs, Microcontrollers and also on recent DSP processors and they have high computational cost, which make them unusable for wearable battery driven health monitoring devices. In this thesis, I have implemented an ECG compression algorithm based on Discrete Wavelet Transform (DWT) and after that I have compressed the DWT coefficients using the Bit Field Preserving (BFP) and Running Length Encoding (RLE) using TMS320C6713 Digital Signal Processor (DSP) using C programming language, Which is a fast Digital Signal Processor (DSP) based on Very Long Instruction Word (VLIW) architecture. This method show efficient realization on TMS320C6713 Digital Signal Processor due to its low complexity , Also this method is simple to implement compared to other compression techniques that requires heavy computations.

The compressed ECG packet obtained from this method can be easily transmitted within the limited payload size of the low power wireless technologies like ZigBee, Bluetooth Low Energy (BLE) etc. This method utilizes a modified DWT-BFP-RLE compression algorithm to dynamically control the size of the compression packet by using the closed loop programming. In this method, to overcome the problem of large compressed packet, which exceeds the payload size, we split the ECG data into smaller blocks and after that we recompress them. In this method, we have applied this dynamic compression algorithm to the small blocks of ECG records which are taken from the MIT-BIH arrhythmia database available at physionet.

1.2 Literature Survey

Szilagyi and David [9] proposed ECG compression using adaptive prediction. In this work they have presented a new ECG compression method, In this method first pre-filtering of ECG signal is done, after that QRS peak detection of ECG signal is done, then R peaks are localized and then the signal is divided into the R-R intervals, then the original signal can be filtered out with less characteristics distortion, this proposed filter is based upon an adaptive long term prediction method, this proposed real-time compression was performed on the ECG samples taken from the MIT-BIH arrhythmia database, this method can reduce the size of the signal at about 50 bits per second without without exceeding 10% Percentage Root mean square Difference (PRD). In this algorithm used entropy coder introduces 10 times less redundancy than an optimized Huffman coder.

Barrera and Ginori [10] have presented a Mean-Shape Vector Quantizer (MSVQ) for ECG signal compression, in this method, they have quantized the mean values of short ECG signal segments as scalars and compressed the single-lead ECG signal by average beat substraction and residual differencing their wave-shapes coded through a vector quantizer, after that an entropy encoder is applied to both vector codes and mean to increase the compression further without degrading the quality of reconstructed signals. The method was assessed through percentage residual difference measurement on the reconstructed signal, also its computation complexity was analyzed considering its real time implementation. This MSVQ method was found to be a very efficient compression method, leading to high compression Ratios (CRs) while maintaining low level of waveform distortion.

Lu, Kim and Pearlman [11] proposed Wavelet compression of ECG signals by the Set Partitioning in Hierarchical Trees Algorithm. Earlier SPIHT algorithm was used for 2D still image coding. They have modified the algorithm for one dimensional signals and applied that to compress the ECG signal, this algorithm was applied on ECG records available at MIT-BIH arrhythmia database and they concluded that this proposed codec was significantly more efficient in compression and computation of ECG signals than the previously proposed ECG compression schemes.This codec also attain the exact bit rate control and also it generates a bit stream progressive in quality.

Zigel, Cohen and Katz [12] have presented ECG signal Compression using Analysis by Synthesis coding, in this they have presented an ECG compression algorithm, called Analysis by Synthesis ECG Compressor (ASEC), this algorithm is based on analysis by synthesis coding and it consists of a beat codebook, long and short term predictors and an adaptive residual quantizer. In this algorithm there is a defined distortion measure to efficiently encode every heartbeat maintaining minimum bit rate and a predetermined distortion level. This proposed algorithm was tested with both PRD and Weighted Diagnostic Distortion (WDD) measure. This algorithm was tested with MIT-BIH Arrhythmia Database records and they got mean compression rate of approximately 100 bits/sec means a compression ratio of 30:1 with reconstruction signal quality (WDD below 4% and PRD below 8%). They have compared this ASEC algorithm with several known ECG compression algorithms and it was found to be superior at all bit rates. They have also applied a Mean Opinion Score (MOS) test, in which the testers were three independent expert cardiologists. In the quantitative test, they have found this ASEC algorithm was found superior than all other tested compression algorithms.

Zigel and Cohen [13] have introduced a Weighted Diagnostic Distortion (WDD) measure for the ECG signal compression, this distortion measure was designed for comparing the distortion between original ECG signal and the reconstructed ECG signal after compression. This WDD was based on PQRST complex diagnostic features such as P wave duration, QT interval, T shape, ST elevation etc. of the original ECG signal and its reconstructed one. Unlike the conventional distortion measure like Percentage Root mean square Difference (PRD), WDD contains direct diagnostic information and therefore it is more meaningful and useful. They have implemented four compression algorithms AZTEC, SAPA2, LTP, ASEC [synthesis coding] in order to evaluate the WDD. A Mean Opinion Score (MOS) test was applied to test the quality of the reconstructed signals and to compare the quality measure, which is also called MOSerror with the proposed WDD measure and the already available PRD measure. The evaluators in the MOS test were three independent expert cardiologists, who studied the reconstructed ECG signals in a blind and a semi-blind manner. They got the correlation between the proposed WDD measure and MOS test measure (MOSerror) better than the correlation between the popular PRD measure and the MOSerror.

Ziya Arnavut [14] presented ECG Signal Compression based on Burrows-Wheeler Transformation and Inversion Ranks of Linear Prediction. This algorithm yield better compression gain in terms of weighted average bit per sample than the recent coding based ECG compression. This work not only not only has better compression, it also has the advantage that with a small modification to it, it can be used as universal coder.

Sahraeian and Fatemizadeh [15] proposed an improved Wavelet-based 2D ECG data compression method using SPIHT [11] and Vector Quantization (VQ) coding, this is a double stage compression technique. In the first stage the Set Partitioning in the Hierarchical Trees (SPIHT) algorithm is used, in the second stage, Vector Quantization (VQ) is applied to the residual image obtained from the first stage. This 2D approach utilizes both the inter-sample and inter-beat redundancies in the ECG signal. This proposed algorithm was applied to several ECG samples from MIT-BIH arrhythmia database and it gives the lower Percentage Root mean square Difference (PRD) than the 1D methods and also several 2D methods for same Compression Ratio.

Fira and Liviu Goras [16] presented a new algorithm for ECG signal compression based on Local extreme extraction, adaptive hysteretic filtering and Lempel-Ziv-Welch (LZW) coding. They have verified this algorithm with the eight of the most frequent, normal and pathological types of cardiac beats and also they have used a Multi-Layer Perception (MLP) neural network (NNs), which is trained with original cardiac patterns and they are tested with reconstructed ones, also the possibility of using Principal Component Analysis (PCA) to the cardiac pattern classification has been investigated, also a new compression measure Quality Score (QS), which is the ratio of Compression Ratio (CR) and the reconstructed error (PRD).

Wu and Hung [17] have presented Reversible Round-Off Non-Recursive Discrete Periodized Wavelet Transform (RRO-NRDPWT) based ECG signal compression using Run Length Encoding (RLE). In this work, they have proposed a Modified Run Length Coding (MRLC) algorithm associated with an efficient quantization scheme for realizing the RRO-

NRDPWT based ECG signal compression system. The MRLC has the characteristics of regularity and low computational complexity, which makes it very suitable for hardware implementation at the cost of compression performance. The proposed algorithm was tested on ECG records from MIT-BIH arrhythmia database and it was found that this scheme was competitive with other wavelet based compression techniques based on the compression performance, also the MRLC improved the traditional Run length coding by about 13%.

Ballesteros and Moreno [18] have proposed FPGA design of ECG compression by using modified convolution scheme of the discrete wavelet transform. In this work a model was developed for a fixed point convolution scheme, which provides good performance in terms of the throughput, the latency, the maximum frequency of operation and also the quality of the compressed signal. The quantization of the coefficients of the filters along with a selected fixed-threshold provide a low error in relation to the clinical applications. This work allows real-time processing of the ECG signal to reduce the redundant information.

Mishra, Thakkar and Modi [19] have proposed ECG Signal Compression using Compressive sensing and wavelet transform, in this they have used Compressive Sensing (CS), which is a novel approach of reconstruction of a sparse signal much below the significant nyquist rate of sampling. In this work, they have done the comparison of the reconstructed 10 ECG signal, which was based on different wavelet families. They have used the fact that ECG signal can be well approximated by the few linear combination of the wavelet basis. They have compared ECG signals by evaluating the performance measures such as Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR), Percentage Root mean square Difference (PRD) and CoC (Correlation Coefficient). They have done reconstruction of the ECG signal by linear optimization process, which uses the sparsity in the wavelet domain. These reconstruction results were analyzed for different compression ratios and they have concluded that the reverse bi-orthogonal wavelet family can give better results for all the values of Compression Ratio as compared to the other families.

Hosseini-Khayat and Ansari-Ram [20] presented ECG signal compression using the Compressed sensing with Nonuniform Binary Matrices. In this work, they have employed compressed sensing in ECG compression. This compression is done in order to minimize the energy consumption during transmission of information from a portable ECG sensor to a server. A non-uniform binary sensing matrix is proposed and implemented to increase the compression ratio and to decrease the distortion in ECG signal. In this better compression ratio and PRD results are obtained as compared to the compressed sensing with uniform binary matrices.

Hun Kim and Yun Kim [21] presented a curvature based ECG signal compression. In their work, they have implemented, an ECG signal compression method for communication on the Wireless Personal Area Network (WPAN), which uses feature points based on the curvatures of the ECG signal. The feature points of P, Q, R, S and T waves, which are the critical components of the ECG signal and have large curvature values compared to other vertexes of the ECG signal, They have extracted these vertexes using a method, which uses the Local maxima of curvatures. They have added extra vertexes according to the Iterative vertex selection method, to minimize the reconstruction errors of the ECG signal. The vertexes selected by this method preserved all the feature points of the ECG signal and it proved to be a very efficient method for the compression of the ECG signal.

Busaidi and Lazhar [22] have designed a low cost microcontroller based health-care device to sense the Electrocardiograph (ECG) signal of heart with the help of two sensing electrodes, an analog amplifier and a low cost microcontroller Atmega32, then they have implemented Digital low-pass, band-pass and band stop filters to filter the ECG signal to reduce the redundancy maintaining acceptable signal quality.

Surekha and Patil [23] have presented ECG signal compression using Hybrid 1D and 2D wavelet transform. In this work, they have combined a Discrete Cosine Transform (DCT) stage with the 1D or 2D Discrete Wavelet Transform. This hybrid stage is very useful for compressing the complete ECG signal as well as the QRS complex of the ECG signal. By using this method, clinically important features of the ECG signal can be properly retained. The QRS complex compression results are better than the complete ECG compression by using this method.(2014)

Zahhad, Hamid and Mohamad [24] have presented a hybrid technique for the compression of ECG signals based on DWT and exploiting the correlation between the signal samples. In this work they have used Discrete Wavelet Transform (DWT) followed by

Differential Pulse Code Modulation (DPCM) and Running Length Coding for the compression of the different parts of the ECG signal. In this method, lossless compression was adopted in clinically relevant parts and the lossy compression was used in those parts, which are not clinically relevant. The proposed algorithm starts by segmenting the ECG signal into its main components (P-waves, QRS-complexes, T-waves, U-waves and the isoelectric waves), after this the resulting waves are grouped into Region of Interest (RoI) and non Region of Interest (nonRoI) parts. After this lossless and lossy compression schemes were applied to the RoI and nonRoI parts respectively. In this method, if a fixed budget of bits is given, then we can spend more bits to represent those parts of the signal that belong to the specific RoI using lossless compression technique and also we can reconstruct these parts with larger fidelity, while allowing other nonRoI parts to suffer larger distortion. For this specific purpose, the correlation between the successive samples of the RoI part was utilized by adopting the DPCM method and the nonRoI part was compressed using DWT, thresholding and coding techniques. The wavelet transformation was used for concentrating the signal energy into a small number of transform coefficients, compression of ECG signal was then achieved by selecting a subset of the most relevant coefficients, which are efficiently coded afterwards. The performance of the proposed algorithm was tested in terms of the compression ratio and the PRD. The results of this algorithm revealed that this technique possesses higher compression ratio and lower PRD in comparition with the other wavelet transformation techniques.

Ukil and Barlocher [25] have proposed implementation of the Discrete Wavelet Transform (DWT) for embedded applications using TMS320VC5510 Digital Signal Processor. In this work, they have presented two fixed-point frameworks for the Discrete Wavelet Transform for real time application of the one-dimentional embedded signal processing. These two were based on the convolution and the lifting scheme respectively, these implementation framework were tested and realized using Texas Instruments 16 bit fixed point processor TMS320VC5510 and the performances of those two frameworks was found to be very promosing towards the development.

Anamika and Vasim [26] have presented ECG compression using Discrete Wavelet transform with coiflets and Daubechies wavelets. In this work, they have performed ECG compression using lossless coding technique, further they have analyzed the results for Coiflets and Daubechies wavelet families based on different parameters like PRD, Compression Ratio and they have concluded that it is much better to use Daubechies wavelets as compared to the Coiflets wavelet for ECG compression, because it gives higher compression ratio and less distortion after reconstruction.

Ranjeet and Anil [27] have presented ECG Signal Compression Algorithm based on Joint-Multi-resolution Analysis (J-MRA). In this work they have proposed a new algorithm based on J-MRA using Gaussian pyramid and wavelet analysis. In signal processing, MRA plays a key role to present signal with reduced number of coefficients with respect to its parental feature. They have implemented this algorithm on ECG signals from MIT-BIH arrhythmia database. This algorithm gives the better Compression Ratio with less distortion as compared to earlier compression techniques.

Grafika and Aprinaldi [28] have proposed ECG signal compression by Predictive coding combined with Set Partitioning in Hierarchical Trees (SPIHT). They utilize the linear prediction between the beats to extract the high correlation between those beats, which can optimize the redundancy between adjacent samples and adjacent beats. In this work predictive coding is used after beat reordering. The predictive coding minimizes the amplitude variance of 2D ECG array resulting in minimized compression error. This method was utilized in ECG samples obtained from MIT-BIH arrhythmia database and it provides the better compression as compared to the original SPIHT algorithm [11] and also it has lower distortion with the same compression ratios compared to other wavelet transformation techniques.

1.3 Motivation

Digital Signal Processors are fast and provide flexibilities to program ECG Signal Processing and many other applications. These Processors overcome many problems which are related with the Analog hardware. These processors are more versatile and reliable than the Analog hardwares. The mathematical methods can be implemented easily using Programs in these processors. Most of these Digital Signal Processors are of low cost and also they consume very less power for operation and they can also provide advanced features like parallel functional units with operating speed of more than 200MHz. In literature survey, we have seen that ECG signal processing can be done on Digital Signal Processors from Texas Instruments. Analog to Digital Converter (ADC) being the limitation of Digital Signal Processing for higher bandwidth signals, with high operating speed of recent ADCs, this limitations are now solved.

The Digital Signal Processor used in this work is TMS320C6713, which is Texas Instrument's C6000 family of processors. This processor consumes very less power and is also of low cost. Real time Digital Signal Processing algorithms can be implemented on this board using C programming, which is high level language and it provide the programmer the flexibility to implement those algorithms with ease. It can also support the assembly language programs, but this limits the portability of the program, because same code cannot be used in any other processor.

1.4 Objective

The objective of this work is implementation of Electrocardiogram (ECG) signal Compression based on Discrete Wavelet Transform (DWT)- Bit Field Preserving (BFP)- Running Length Encoding (RLE) method using a C6713 Digital Signal Processor to obtain satisfactory compression performance with less reconstruction error and to realize fully operating ECG signal Processor within the limits set by this C6713 DSP hardware and thus paving the way to implement more complex algorithms to be implemented on this DSP board.

1.5 Organization of the Thesis

Chapter 1: Introduction

In this chapter the overview of ECG signal along with Discrete Wavelet transform is presented, also some compression techniques, which can be used to compress the ECG signal. Some literature regarding various techniques for ECG signal processing, its compression and its processing on Digital Signal Processors has also been presented. Then some advantages of Digital Signal Processor over Analog processors has also been discussed. Then at last the objective of the current work has been summarized.

Chapter 2: The TMS320C6713 Digital Signal Processor

In this chapter first basic architecture of the Texas Instruments TMS320C6713 Digital Signal Processor is presented. Then its functional units, Input/Output operation for real time signal processing has been studied, then software used by the processor for the programming called Code Composer Studio (CCS) along with various support files and its various abilities has also been studied.

Chapter 3: Study of ECG Signal, DWT and Various Compression Techniques

In this chapter, first basics of ECG signal has been discussed, then the details of the Discrete Wavelet Transform (DWT) has been presented, then various types of compression methods have been discussed. Also comparison between BFP-RLE technique and other techniques have been discussed.

Chapter 4: Implementation of the Compressed ECG Signal

In this chapter, the implementation of ECG signal compression based in the DWT-BFP-RLE algorithm on TMS320C6713 Digital Signal Processor has been discussed. This chapter introduces the algorithm for the proposed work. Also in this chapter compression results for various ECG records have also been presented.

Chapter 5: Conclusion and Future Scope of the Algorithm

In this chapter, the conclusion drawn from the work along with the Future Scope of the proposed work has been discussed.

Chapter 2

The TMS320C6713 Digital Signal Processor

2.1 Introduction

Digital signal processors such as the TMS320 family of processors, are used in wide range of applications such as in communication, control, speech processing and so on. They utilize the hardware resources along with the current technology to give us high performance hardware devices in real time. The Texas Instruments TMS320C6713 Digital Signal Processing Starter Kits (DSK) is a low cost development platform for real time digital signal processing applications and it can be used to implement marketable version of a specific signal processing operation. Code Composer Studio (CCS) v3.1 is used in order to enable programs written in C or assembly language to be compiled and/or assembled, linked and downloaded to run on the DSK. This board along with CCS is utilized to implement the Discrete Wavelet Transform based ECG signal compression.

In section 2.2 basic architecture of the DSK along with various functional units, Regieters, Memory Organization and On-chip Peripherals is explained. In section 2.3 Input/Output operation of the DSK along with AIC23 codec, McBSP and Interrupts have been explained. In section 2.4 Code Composer Studio with various support file has been discussed. In section 2.5 summary of the chapter is given.

2.2 TMS320C6713 Digital Signal Processor Architecture

The TMS320C6713 onboard the DSK is a floating point processor based on the Texas Instrument's VelociTI^{TM} architecture, an advanced Very-Long-Instruction-Word (VLIW) architecture [29] for DSPs. It has an architecture based on the real time signal processing, in which many functional units are running in parallel to achieve high performance by Instruction level parallelism [30]. Advanced features of this architecture include instruction packing, conditional branching and pre-fetched branching, all of these overcome problems that were associated with previous VLIW implementations. The architecture is highly deterministic, with few restrictions on how or when instructions are fetched, executed, or stored. This architectural flexibility is key to the breakthrough efficiency levels of the C6000 compiler.

The floating point architecture present in C6713 DSP is more expensive and has additional complex circuitry for floating point operation support that increases the chip area. Floating point architecture eliminates the burden of the programmer by eliminating the constant need for scaling and overflow handling of the data which is constantly required in a fixed point processor. It also increases the dynamic range of the data being used in the computation. The TMS320C6713 DSP board is capable of both fixed point and floating point operations and it utilizes the modified Harvard architecture [31], in which program and data memories are stored in different memory spaces, so that a single clock cycle is required to fetch both instructions and data, unlike the old Von Neumann architecture [32] in which both program and data memory are stored in the same memory space. So modified Harvard architecture is much faster than the old Von Neumann architecture.

Figure 2.1,[33] shows the functional block diagram of board with on chip peripherals and cache memory structure. The operating frequency of the processor is 225MHz, Thus the processor can process information at a rate of 1.35 Giga Floating Point Operations Per Second (GFLOPS) [33], 1800 Million Instruction Per Second (MIPS) [34], and with dual fixed/floating point multipliers up to 450 million multiply-accumulate operations per second (MMACS) [34] with a 4.44 ns instruction cycle time.

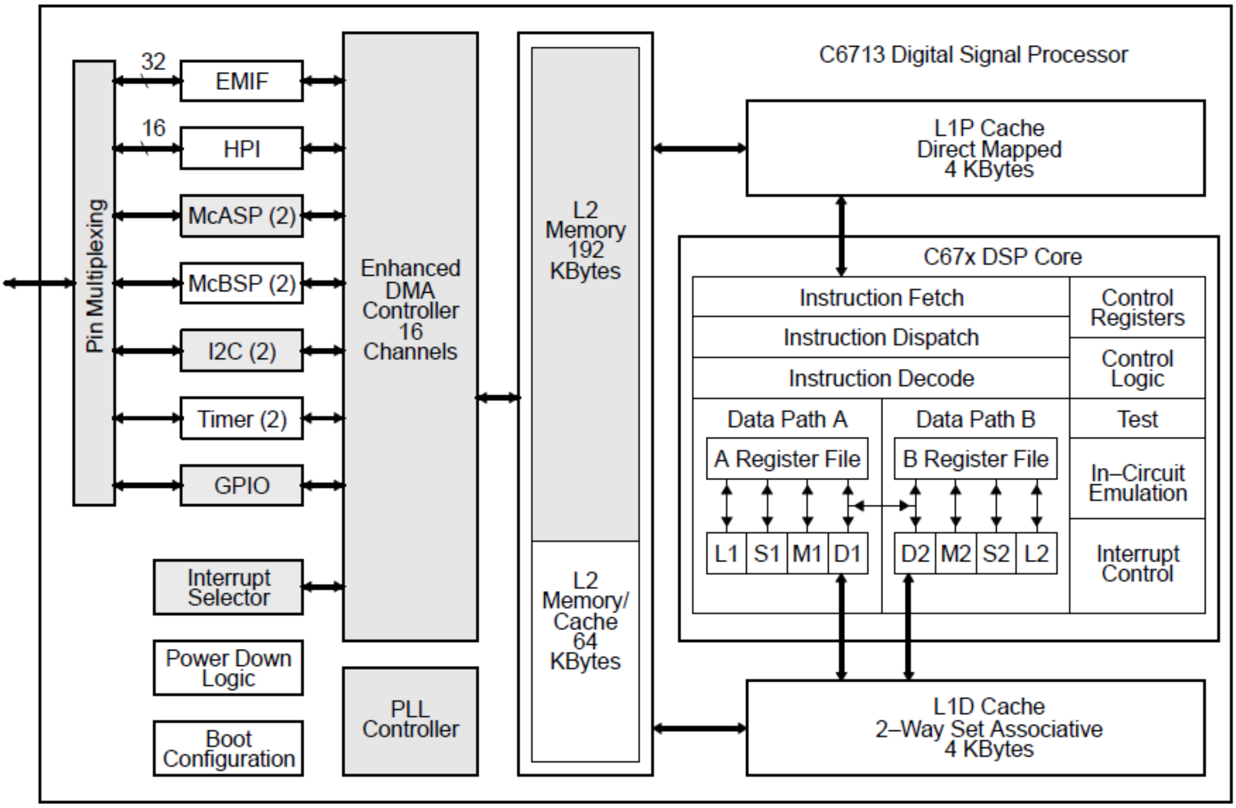

Figure 2.1: Functional Block Diagram of TMS320C6713. (Courtesy of Texas Instruments)

2.2.1 Functional Units

The CPU consist of 8 independent functional units [29] divided into two data paths, A and B [29] as shown in figure 2.1, both paths A and B have equal number of functional units, which can perform equal and simultaneous operations, this is the cause of increase of speed and Instruction level parallelism in the CPU. Each path has a unit for Multiply operations (.M), for Logical and Arithmetic operations (.L), for branch, bit manipulation and arithmetic operations (.S), and for loading/storing and arithmetic operations (.D), all of these can perform single precision or double precision floating point operations. The .S and .L units are for arithmetic, logical and branch instructions. All data transfer makes use of the .D units.

The arithmetic operations, such as subtract or add (SUB or ADD), can be performed by all the units, except the .M units (one from each data path), which are specific for single cycle multiplications, which form the basis for faster Multiply And Accumulation (MAC) operations. The eight functional units consists of four floating/ fixed point ALUs (two .L and two .S), two fixed point ALUs(.D units), and two floating/fixed point multipliers (.M units).

Each functional unit paths A and B includes a set of sixteen 32 bit registers, A0 through A15 for path A and B0 through B15 for path B as shown in figure 2.1. Each functional unit can read directly from or write directly to the register file within its own path. As shown in figure 2.1, (.L1, .S1, .M1, .D1) can communicate with the register files of A path, similarly (.L2, .S2, .M2, .D2) can communicate with the register files of B path. Two cross paths allow functional units from one data path to access a 32-bit operand from the register file on the opposite side, i.e. the functional units on the one side can access the register file from the other side using a cross path[29].

2.2.2 Fetch and Execute Packets and Parallelism

Pipelining is the key feature of the 6713 DSP, there is a three stage pipelining structure[29] in this architecture, this structure provide the advantage of rearranging the group of instructions that can be executed in parallel within the same cycle time in to parallel groups of Execute Packets (EP) [29]. The number of EPs in a Fetch Packet (FP) can vary from one (with eight parallel instructions) to eight (with no parallel instructions). The VLIW architecture was modified to allow more than one EP to be included within an FP [29,35].

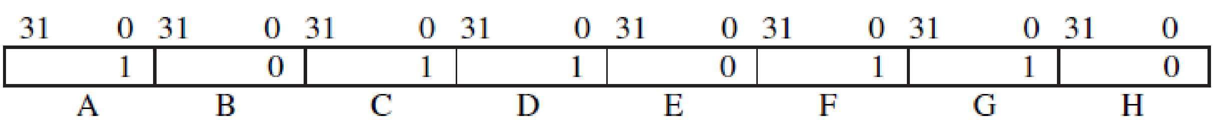

Figure 2.2: One Fetch Packet (FP) with three Execute Packets (EPs)

The figure 2.2 [35] shows a fetch packet of 8 instruction containing 3 execute packets, which means 3 groups of instructions are formed, where within each group, the instructions are parallel. The least significant bit of every 32 bit instruction is used to determine if the next or subsequent instruction belongs in the same EP (if 1) or is the part of the next EP (if 0).

Let us consider an FP with three EPs: EP1 with two parallel instructions Instruction A ‖ Instruction B, EP2 with three parallel instructions Instruction C ‖ Instruction D ‖ Instruction E and EP3contains three parallel instructions Instruction F ‖ Instruction G ‖ Instruction H. The FP for this would be as shown in figure 2.2. Bit 0 (LSB) of each 32-bit instruction contains a ‘p’ bit that signals whether it is parallel with a subsequent instruction. For example the ‘p’ bit of instruction B is zero, denoting that it is not within the same EP as the subsequent instruction C. Similarly instruction E is not within the same EP as instruction F [7].

2.2.3 Registers

There are two sets of register files, each set with 16 registers, are available: register file A (A0 through A15) and register file B (B0 through B15) [35]. It uses the load-store architecture [36], In this type of architecture, operands required by the functional units are not directly accessed from the external memory. The operands and data are pre-fetched by the by the data (.D) functional units and stored in the registers. As the access to register do not require any addressing mechanism like accessing an external memory, it works faster. This provides a faster approach to calculation where data is available without wasting many clock cycles over proper signal and chip enable functions.

Registers A0, A1, B0, B1 and B2 are used as conditional registers. Registers A4 through A7 and B4 through B7 are used for circular addressing. Register A0 through A9 and B0 through B9 (except B3) are temporary registers. Any of the registers A10 through A15 and B10 through B15 used are saved and later restored before returning from a subroutine [35].

A 40-bit data value can be contained across a register pair can be contained across a register pair. The 32 least significant bits (LSBs) are stored in the even registers (e.g., A2) and the remaining 8 bits are stored in the 8 LSBs of next-upper (odd) register (A3). A similar scheme is used to hold a 64-bit double precision value within a pair of registers (even and odd). All these 32 registers are considered as general purpose registers [35].

2.2.4 Memory Organization and Bus Structure

TMS320C6713 DSP’s Internal memory includes a two-level cache architecture [37] with 4kB of level 1 program cache (L1P) and 4kB of level 1 data cache (L1D) and 256kB of level 2 memory shared between program and data space as shown in figure 2.1. 64kB of shared memory space may be configured as a mapped memory or cache or a combination of the two. The remaining 192kB of L2 memory is usually designated as mapped SRAM. It also has a glue-less (direct) interface to both synchronous memories (SDRAM and SBSRAM) and asynchronous memories (SRAM and EPROM) [35].

Internal buses include a 32-bit program address bus, a 256-bit program data bus to accommodate eight 32-bit instructions in a packed form. It also has two 32-bit data address buses, two 64-bit data buses and two 64-bit store data buses. With a 32-bit address bus total memory space is 2^{32}= 4GB including four external memory spaces: CE0, CE1, CE2, CE3.

Table 2.1 [37,38] shows the memory map for C6713 DSK. It includes 264kB of internal memory, which starts at 0×00000000, and 16MB of external SDRAM, mapped through CE0 starting at 0×80000000. It also includes 512kB of flash memory (out of which 256kB is readily available for user), mapped through CE1 starting at 0×90000000, CPLD also share CE1 along with flash and it starts at 0×90080000. Daughter Cards have also provided the address spaces and are mapped with CE2 and CE3.

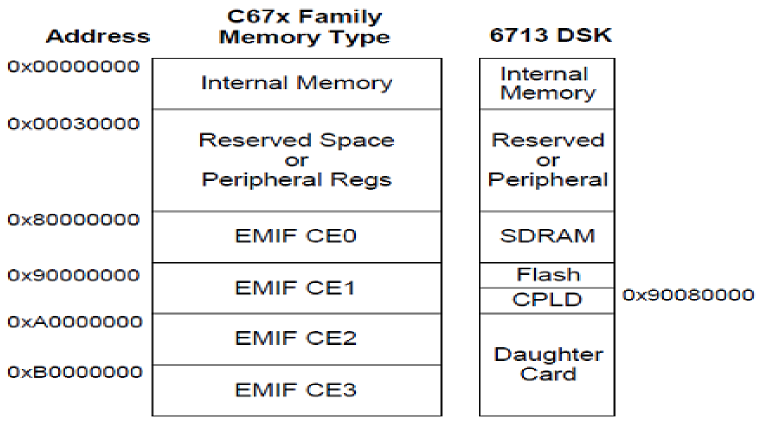

Table 2.1: Memory Map for C6713 DSK

The connection to the external memory is through the External Memory Interface (EMIF) and they are controlled by peripheral registers and hence their operation is faster.

2.2.5 On-Chip Peripherals

There are many on-chip peripherals [30], which serve various functions. They are :

1) **Timers:** Two 32 bit timers can be used to time and count events and to interrupt the CPU. A timer can direct an external ADC to start conversion or a DMA controller to start a data transfer.

2) **McBSP:** Two Multichannel Buffered Synchronous Ports (McBSP) are available. They provide an interface to external peripherals and have direct interface to AIC23 codec. McBSPs have features such as full duplex communication, independent clocking and framing for receiving and transmitting. They also allow data sizes between 8 and 32 bits.

3) **DMA:** Direct Memory Access (DMA) allows for the transfer of data to and from the internal memory or external devices without intervention from the CPU. Sixteen Enhanced DMA channels (EDMA) can be configured independently for data transfer. DMA can access on-chip memory and the EMIF, as well as HPI. Data of different sizes can be transferred : 8-bit bytes, 16-bit half words, and 32-bit words.

4) **McASP:** Multichannel Audio Synchronous Ports are similar to the McBSP but they are specific for Audio applications. They can handle a maximum frequency output of 192 kHz stereo signal using the Philips Inter-IC Sound (I2S) format.

5) **HPI:** Host Port Interface provides a 16 bit parallel interface to a host. The host fuctions as a master and can access the entire memory map of the DSP.

6) **EMIF:** External Memory Interface provides a glueless interface to variety of external devices including SDRAM, SBSRAM, ROM etc. so extra memory other than on-chip memory may also be available.

7) **Power Down Logic:** The Power Down Logic allows reduced clocking to reduce power consumption. Most of the power is dissipated during switching of CMOS circuits from one state to another. It prevents the some or all of the chip's logic from switching.

2.3 Input and Output with the TMS320C6713 DSK

Like basic DSP systems, TMS320C6713 DSK comprises a Digital Signal Processor and Analog interfaces as shown in Figure 2.3 [35], it is also suitable for processing audio frequency signals. TMS320C6713 DSK provides such a system using TMS320C6713 (C6713) floating point processor and the TLV320AIC23 (AIC 23) codec [43], here codec refers to the coding of analog waveform as digital signals and decoding of digital signals as analog waveforms. AIC23 code performs both the analog-to-digital conversion (ADC) and digital-to-analog conversion (DAC) functions.

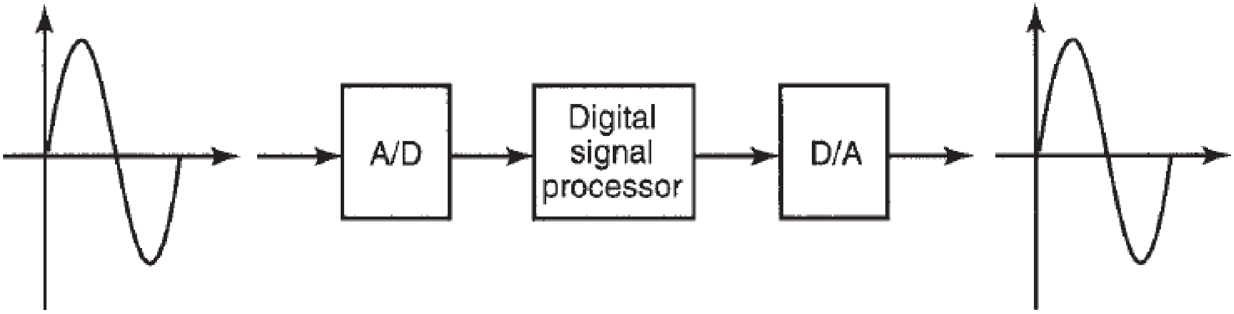

Figure 2.3: Basic Digital Signal Processing systems with Input and Output

2.3.1 TLV320AIC23 (AIC23) ONBOARD STEREO CODEC

The AIC23 codec forms the interface between C6713 DSK and the external world. Figure 2.4 [38] shows the AIC23 codec on TMS320C6713 DSK.

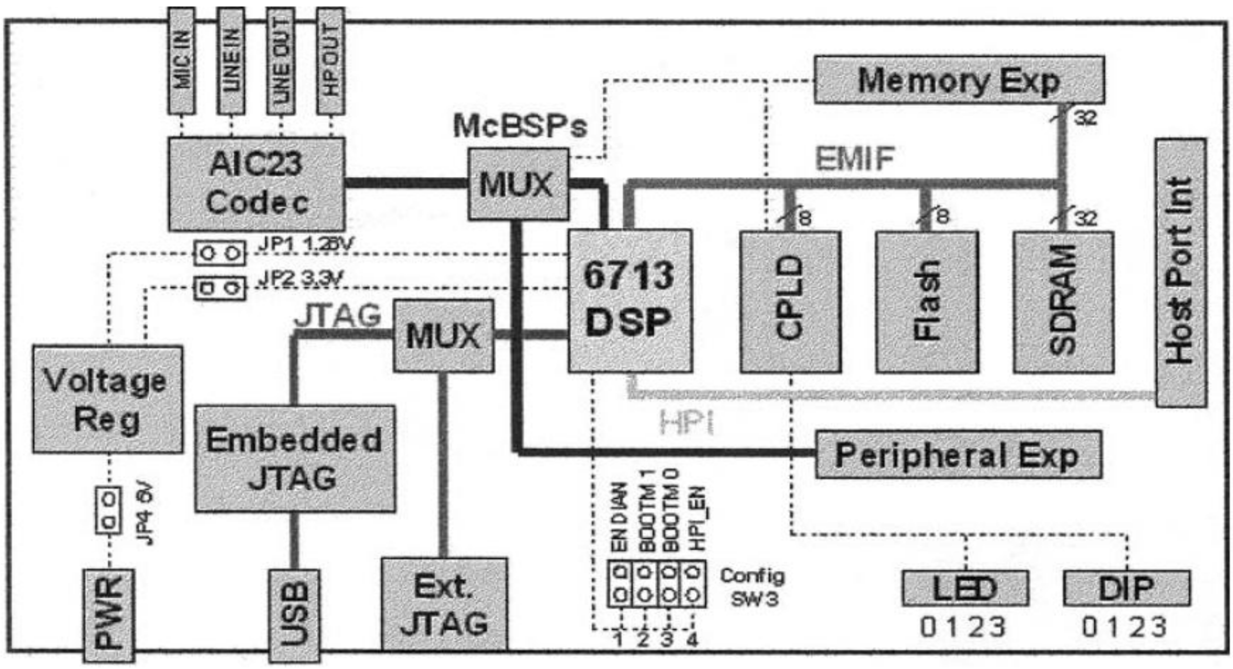

Figure 2.4: Block Diagram of TMS320C6713 DSK with AIC23 codec (Courtesy of Texas Instruments)

The TLV320AIC23 codec is a stereo codec with integrated functionality of headset and microphone. The analog to digital converter (ADC), or coder part of the codec converts the analog signal into a sequence of sample values (16 bit signed integer) to be processed by the digital signal processor. The digital to analog converter (ADC) or the decoder part of the codec reconstructs an analog output signal from a sequence of sample values (16 bit signed integer) that have been processed by the digital signal processor.

AIC23 codec is based on the sigma delta technology [43]. This codec allows high quality playback and at most 90dBA signal-to-noise ratio [43]. The AIC23 codec has inputs as LINE IN for use with CRO and MIC IN for microphone input purposes. The outputs of the codec are LINE OUT and HEADPHONE OUT. All the ports are stereo channel. The LINE and the MIC inputs are multiplexed and only one of them can be used at a time.

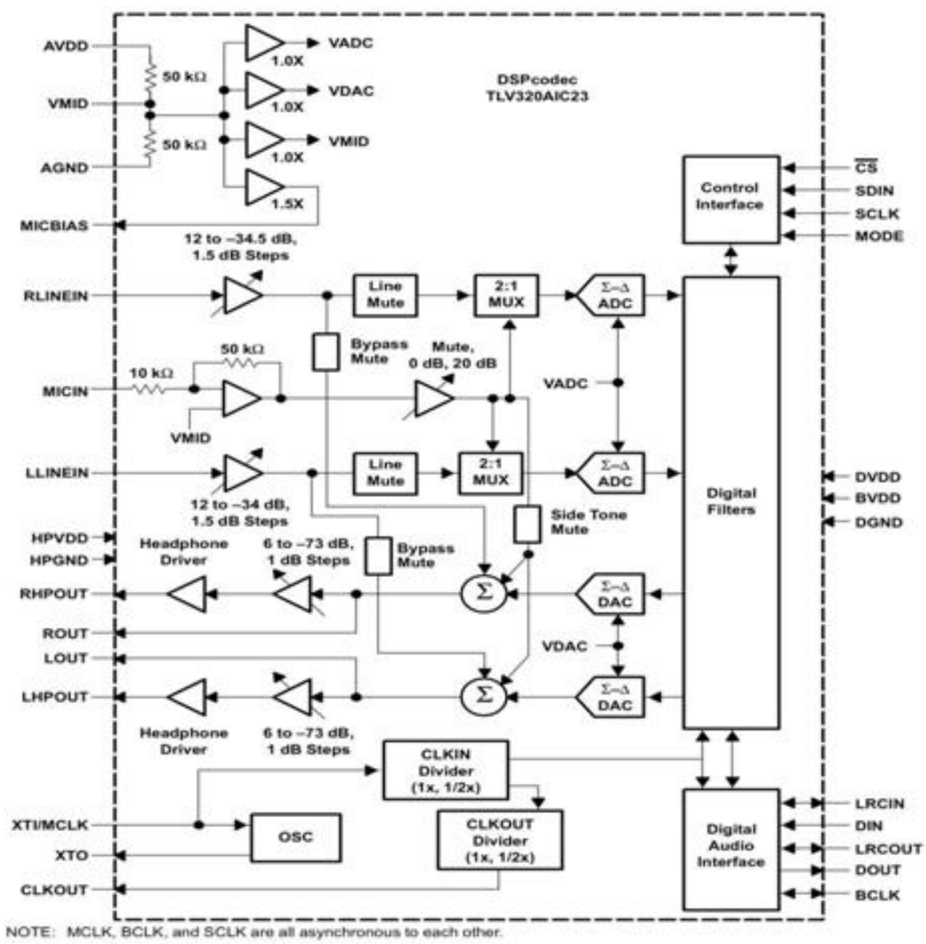

Figure 2.5: Functional Block Diagram of TLV320AIC23 Codec.

Figure 2.5 shows the functional block diagram of AIC23 codec [43]. A 12MHz crystal supplies the clock to the AIC23 codec. Using this master clock with oversampling rates of 250Fs and 272Fs, exact audio sample rate of 48kHz (12MHz/250) and the CD rate of 44.1kHz (12MHz/272) can be achieved. The sampling rates of AIC23 can be configured to be 8, 16, 24, 32, 44.1, 48, 96 kHz [35].

For input and output operations AIC23 codec uses two Multichannel Buffered Serial Ports (McBSP) on the C6713 named McBSP0 and McBSP1. McBSP0 is used as a unidirectional channel to send a 16-bit control word to AIC23 codec. McBSP1 is used as a bidirectional channel to send and receive audio data. The codec can be configured for data transfer word-lengths of 16, 20, 24 or 32 bits [35].

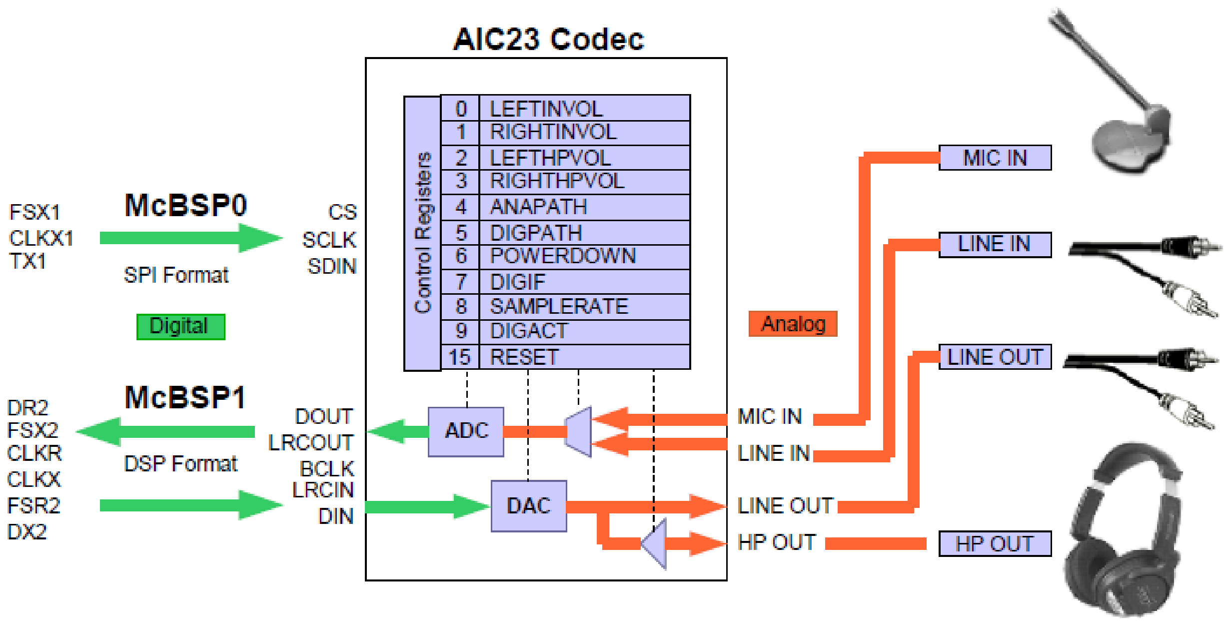

Figure 2.6: TMS320C6713 DSK AIC23 CODEC Interface

Figure 2.6 shows the block diagram of AIC23 interfacing with McBSP and the outside world. The LINE IN and HEADPHONE OUT signal paths within the coded contain configurable gain elements with ranges of 12 to -34dB in steps of 1.5dB, and 6 to -73dB in steps of 1dB, respectively. The maximum allowable input signal level at the LINE IN inputs to the codec is 1V rms. However C6713 contain a potential divider circuit with gain of 0.5 between their LINE IN sockets and the codec itself with the effect that the maximum allowable input signal level at the LINE IN sockets on the DSK is 2Vrms, Above this level input signals will be distorted.

2.3.2 Multichannel Buffered Serial Ports (McBSP)

There are two McBSPs available with this board. They provide an interface to the in expensive external peripherals. They have features like full duplex communication, independent clocking and framing for receiving and transmitting [30,35]. McBSPs allow several data sizes between 8 to 32 bits. External data communication can be achieved while data are being moved internally.

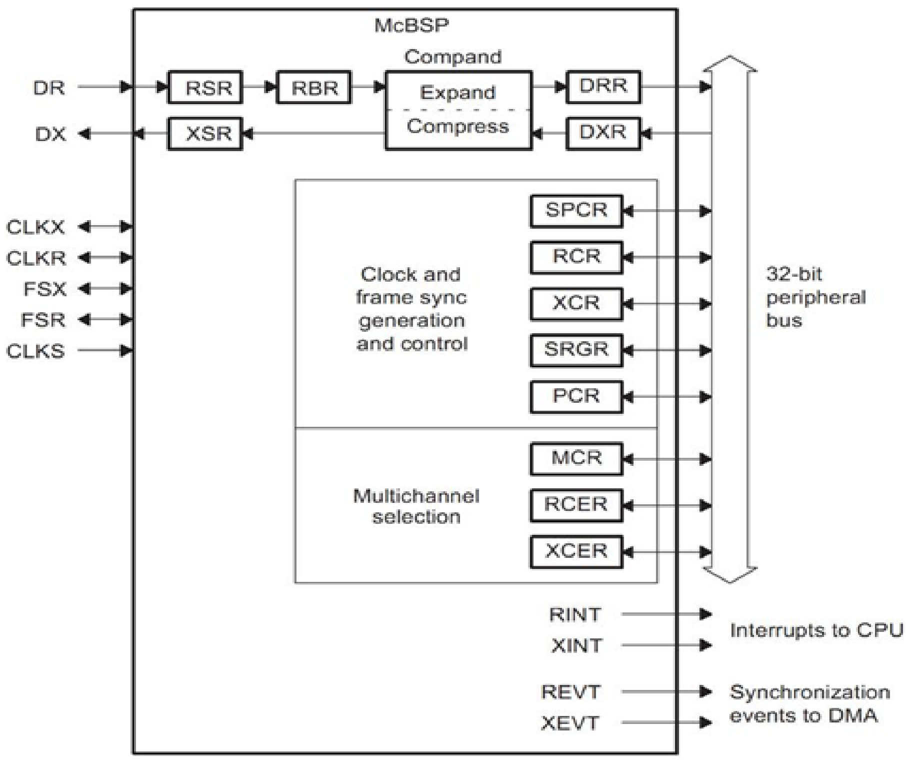

Figure 2.7: Internal block diagram of McBSP. (Courtesy of Texas Instruments)

Figure 2.7 [30] shows the internal block diagram of a McBSP. The data transmit (DX) and data receive (DR) pins are used for data communication. CLKX, CLKR, FSX and FSR are used for control information (clocking and frame synchronization). The CPU or DMA controller reads the data from the Data Receive Register (DRR) and writes data to be transmitted to the Data Transmit Register (DXR). The Transmit Shift Register (XSR) shifts this data to DX. The Receive Shift Register (RSR) copies the data received on DR to the Receive Buffer Register (RBR). The data in the RBR are then copied to the DRR to be read by the CPU or the DMA controller.

Other registers like the Serial Port control register (SPCR), receive/transmit Control Register (RCR/XCR), receive/transmit channel enable register (RCER/XCER), pin control register (PCR) and sample rate generator register (SRGR) are used for further data communication [35].

The two McBSPs are used for input and output through the on-board codec. McBSP0 is used for control and McBSP1 is used for transmitting and receiving data [37]. The left input sample is sent first, after that the right input sample is sent. Thus if a frame of 32-bit word data is configured, then the upper 16-bits forms the left channel input and the lower 16-bit form the right channel input. The codec generates a frame synchronization as soon as it send theleft channel input. The reverse process of sending out data in such a packed format is done, therefore the program operating the codec must take care of the synchronized left and right channel packing of data for proper operation.

McBSP can be controlled by C functions provided by TI Chip Support Library (CSL). The TI's chip support library(CSL) guide [44] provides all the C functions and macros required to control all the peripherals, including McBSP of 6713 DSP. The basic transmitter and receiver operations of McBSP are as follows:

Transmitter Operation:

- The CPU writes a 32-bit word into data transmit register (DXR), during this XRDY flag is reset.
- As the data writing is complete the Transmit Frame Synch (FSX) is set and shift of data take place from the transmit shift register (XSR). This shift of data from XSR causes a similar shift of data from DXR to XSR in parallel. As this transfer is complete, the DXR is now empty and XRDY flag is set to high indicating there is space to accept more data.
- As soon as the XRDY flag is set from 0 to 1, an interrupt request XINT is sent to the CPU and also event notice XEVT is sent to the EDMA, which is used when EDMA is used for transmitting and receiving.

Receiver Operation:

- In receiver side first data is sent from the McBSP to the CPU. Receiver operation is exactly the reverse of the transmitter operation. In this case the Receive Frame Synch (FSR) is set high and data is shifted serially into Receive Shift Register (RSR).
- If a predefined number of bits of data, say 32 bits is transferred, the bits from RSR are shifted in parallel to Receive Buffer Register (RBR) if it is empty. The data from RBR is copied into the Data Receive Register (DRR) if it is empty.
- The availability of the data in the DRR is indicated by the RRDY flag in the SPCR, when the data from RBR is shifted to the DRR, the RRDY flag is set to 1 indicating that buffer is full and the CPU/ EDMA may receive the data.
- When the RRDY flag goes from 0 to 1, it triggers an interrupt request RINT for the CPU, also an event notice REVT is also sent to EDMA.

2.3.3 Interrupts of TMS320C6713

An interrupt is an event that stops the current process in the CPU, so that the CPU can attend to the task needing completion because of the event. This interrupts can be available on chip or off chip, such as timers, analog to digital converters (ADC) or other peripherals. When an interrupt is triggered the contents of the registers are saved and the processing continues to an Interrupt Service Routine (ISR), after that contents of the registers are restored and normal execution continues [35].

There are three types of interrupts on the CPU of the C6713 DSP. The reset interrupt has the highest priority and corresponds to the $\overline{\text{RESET}}$ signal. The non-maskable interrupts is the interrupt of second highest priority and corresponds to the NMI signal. The lowest priority interrupts are interrupts 4-15. They correspond to INT4-INT15 signals. There are total 16 interrupt sources. They include two timer interrupts, four external interrupts, four McBSP interrupts and four DMA interrupts. INT4-INT15 are CPU interrupts. An Interrupt selector is used to choose among these 12 interrupts.

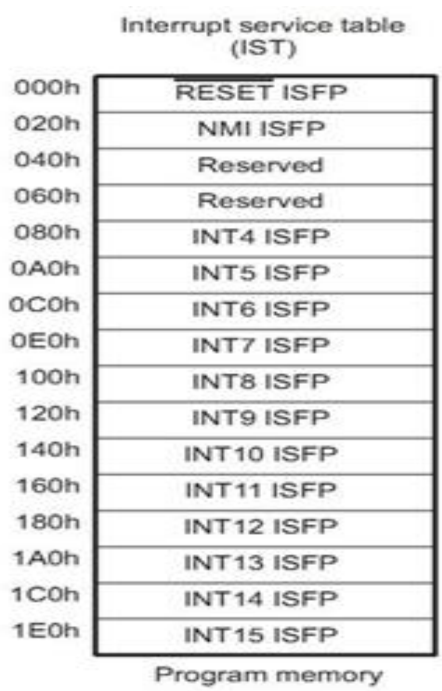

Figure 2.8: Interrupt Service Table with decreasing priority of Interrupts

Figure 2.8 [29] shows the address and contents of Interrupt Service Table (IST). When the CPU begins the processing of an interrupt, it references the Interrupt Service Table (IST). The IST is a table of fetch packets that contain code for servicing the interrupts. The IST consists of 16 consecutive fetch packets. Each interrupt service fetch packet (ISFP) contains eight instructions. A simple interrupt service routine may fit in an individual fetch packet.

Interrupts INT4-INT15 are user interrupts and they are programmable. There are extensions provide by TI to write our own Interrupt Service Routine (ISR), and use that ISR with a specific interrupt, so that when that interrupt is triggered that ISR is executed, this extension requires writing a function, where the function declaration starts with 'interrupt' keyword. The format of TI's extension is as follows [39],

Interrupt void isr_func () {…}

The 'isr_func' above can be anything of a user's choice, but the same name must be hooked to a certain CPU interrupt defined above. This hooking process and setting of the required interrupt control register to accomplish a fully working ISR is given in [37] and the details of the interrupt control register and their settings are available in [29].

The basic operation that is utilized for real time data transfer using Interrupts are as follows,

- First initialize the AIC23 codec, the McBSP0 and the McBSP1 with suitable sampling frequency, say 8 kHz as in our implementation.

- The CPU interrupt say INT11, in our implementation, to the Transmit Interrupt Request XINT1 of McBSP1. Then CPU interrupt INT11 must be mapped to the ISR selected by us.

- After all the hooking process, Interrupts are required to be enabled.

- When the program is executed, it is required that the program must be set into an Infinite and interruptible loop. The interrupt must be triggered in regular intervals and input/output of the data is done as per the interrupt inside the ISR. When ISR is complete the normal execution of the CPU is continued.

2.3.4 Polling versus Interrupt Driven Input/ Output

If the data going out of the CPU, CPU keeps checking of the XRDY flag in Polling method and similarly if the data is going in opposite direction, then CPU keeps track of the RRDY flag. If the data is being received by the CPU, CPU constantly polls the RRDY flag to be set to ‘1’. The RRDY flag is set to ‘1’ when the DRR flag is full of data and when the CPU detects that, it takes out the data from DRR. Now RRDY flag is reset to ‘0’ , when DRR is empty. If the data is transferred out of the CPU, the CPU checks the XRDY flag to be set as ‘1’. If it is ‘1’ then it indicates that the DXR is empty and it means that data from the CPU can be stored, when CPU detects that DXR is empty, it dumps the data into the DXR resulting XRDY flag is reset to ‘0’, which indicate that the DXR is full.

In polling method, CPU continuously check the flags using Application Programming Interface (API) functions provided by the Texas Instruments [16]. In polling method there is waste of CPU cycles due to continuously waiting by the CPU. In case of Interrupt Driven Method the change of level of the RRDY flag and the XRDY flag causes an interrupt to generate, this interrupt is linked to the one of the programmable interrupts INT4-INT15 of the CPU. If this interrupt is triggered, an ISR is used, which is designed in C to utilize the McBSP functions for data input/output operation. By Interrupt driven method waiting overhead can be avoided. In this method, during the processing of Interrupt, normal CPU operation is halted.

2.4 Code Composer Studio

Code Composer Studio (CCS) provides an Integrated Development Environment (IDE) for the implementation of the real-time digital signal processing applications based on C programming language [35,42]. It incorporates a C compiler, an assembler and a linker. It has graphical capabilities and also supports the real time debugging.

We give the program with .c extension in our project. First of all the C compiler compiles this C source program with .c extension to produce an assembly language source file with extension .asm. The assembler assembles this .asm file to produce a machine language object file with extension .obj. The linker combines this object file along with the object libraries as input to produce an executable file with extension .out. This executable file can be loaded and run directly on the digital signal processor.

A Code Composer Studio project comprises all of the files required to generate an executable file. In CCS options for adding or removing different types of files are available. Also information about exactly how these files are to be used to generate an executable files are also provided. In CCS there are a number of debugging features are available, including setting breakpoints and watching variables, viewing memory, registers and mixed C and assembly code, graphing results and monitoring execution time.

Real time analysis can be performed using CCS's Real-Time Data Exchange (RTDX) facility. RTDX allows data exchange between the host PC and the target DSK as well as analysis in real-time without halting the target DSK.

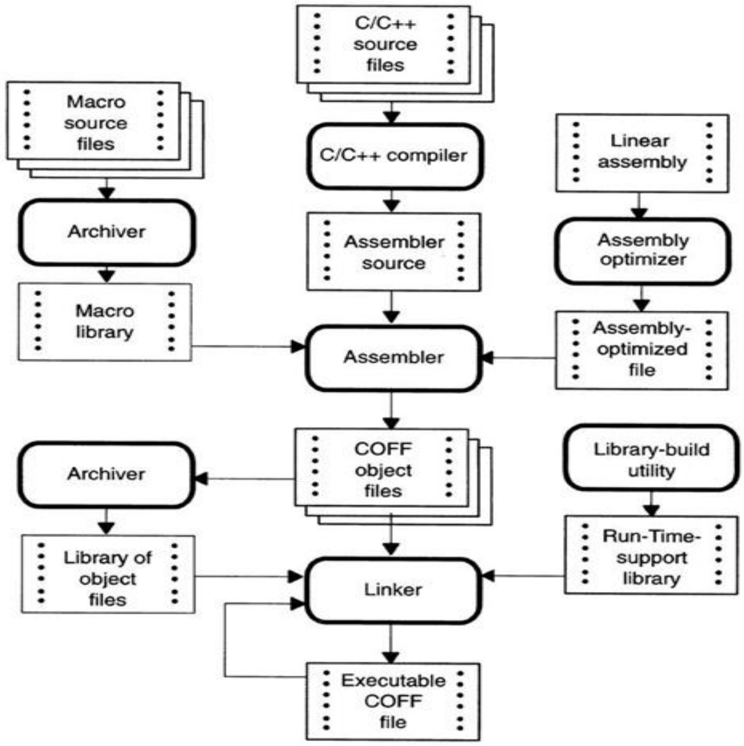

Figure 2.9: Building Programs using CCS (Courtesy of Texas Instruments)

Figure 2.9 [37] shows the process of creation and building of a program using CCS. CCS has error detection capabilities and also syntax highlighting capabilities, which makes it more powerful. If an error is detected in program, then by clicking that error in the error window, we can directly jump to that location of the error in the source file.

The C compiler provides several levels of optimization [41], which increases the efficiency of the code. The compiler also automatically schedules the source code, so that the functional units can be executed in parallel.

CCS also provides the facility to calculate the amount of CPU cycles required by a certain code. It also provide graphical capabilities, The CCS can plot the graph of any output if the data or coefficients related to that output are given. CCS allows hand optimization of code by examine and re-writing the source code that is performing in an un-optimized manner. CCS also provides statistics about the execution times of the program portions. It is required that the program creation time should be as low as possible, so entire programs are written in C language.

C programs are preferred over assembly language programs because C programs can be ported to another system with suitable C compiler present in that system, whereas Assembly language programs are architecture specific. Also using C programs readability can be increased.

CCS can be used to build programs using DSP/BIOS [45], which is a real-time operating system. During execution CCS can read files using the probe points, but to do this, it must halt the CPU to read the data, which causes loss of important CPU cycles. By using DSP/BIOS operating system, this problem can be solved because using this input/output from the files can be done while the CPU is still in the operation.

2.4.1 Support Files

- **Initialization/Communication file (c6713dskinit.c):** It contains the definition of a number of functions used to initialize the DSK.

- **Header File (c6713dskinit.h):** It contains the function prototypes as well as initial settings for the control register for AIC23 codec.

- **Vector Files (vectors_intr.asm, vectors_poll.asm):** These are assembly language files. Vectors_intr.asm is used in interrupt driven programs. It is used to set up the Interrupt Service Routine (ISR). Vectors_poll.asm is used in polling based programs.

- **Source Files:** This files are the actual programs that we run. They may be written in C language or in assembly language.

- **Library Files:** These are the support files required to access the functionalities of the peripherals. They provide functions and macros to control interrupts, peripherals etc. there are three most important library files which I have used in my project are,

a) **Board Support Library (dsk6713bsl.lib):** It contains the C language functions for controlling the onboard external peripherals on the C6713 DSK.

b) **Chip Support Library (csl6713.lib):** It contains the C functions and macros required for controlling the on-chip peripherals and the CPU interrupt controller.

c) **Run Time Support Library (rts6700.lib):** It is required by the linker for unresolved references such as identifying the "main()" function in C.

- **Linker Command File (c6713dsk.cmd):** It specifies the memory configuration of the internal and external memory available on the DSK [35] and the mapping of sections of code and data to absolute addresses in that memory. For example, the .text section produced by the C compiler, is mapped into IRAM, i.e. the internal memory of C6713 DSP, starting at the address 0×00000220. The section .vectors created by vectors_intr.asm or by vectors_poll.asm is mapped into IVECS, i.e. the internal memory starting at address 0×00000000 (the interrupt service table).

```
MEMORY
{
  IVECS:    org=0h,          len=0x220
  IRAM:     org=0x00000220, len=0x0002FDE0 /*internal memory*/
  SDRAM:    org=0x80000000, len=0x00100000 /*external memory*/
  FLASH:    org=0x90000000, len=0x00020000 /*flash memory*/
}
SECTIONS
{
  .EXT_RAM :> SDRAM
  .vectors :> IVECS      /*in vector file*/
  .text    :> IRAM
  .bss     :> IRAM
  .cinit   :> IRAM
  .stack   :> IRAM
  .sysmem  :> IRAM
  .const   :> IRAM
  .switch  :> IRAM
  .far     :> IRAM
  .cio     :> IRAM
  .csldata :> IRAM
}
```

Figure 2.10: A sample linker command file.

Figure 2.10 [35] shows a sample linker command file, which is divided into memory and sections. The 'MEMORY' portion defines the starting and the ending addresses for the different memory elements of the CPU and peripherals as the memory mapped devices. The ‘SECTIONS’ portion is required to logically divide the memory into the portions of data, program, global variables, stack etc.

2.5 Summary

This chapter gives the brief architecture, the peripherals and the input/output operations of TMS320C6713 digital signal processor along with Code Composer Studio provided by Texas Instruments to program the DSP’s CPU, also we have seen various support files required by the CCS to program the CPU. In this work this board is used by me to implement the compressed ECG signal.

Chapter 3

Study of ECG signal, DWT and Various Compression Techniques

3.1 Introduction

The Electrocardiogram (ECG) is one of the most important signal used by the medical practitioners to analyze the electrical events in the human cardiac cycle. An efficient ECG compression technique can reduce the payload of the real time ECG transmission as well as it can reduce the amount of data storage in long-term ECG recording. Discrete Wavelet Transform (DWT) is an appropriate tool for analysis of ECG signals as it removes the shortcomings of Fourier and other transforms. There are various DWT based compression techniques, which can be used like Vector Quantization (VQ) [15], Set Partitioning in Hierarchical Trees Algorithm (SPIHT) [11], Lifting based scheme [64], and Running Length Encoding schemes etc. In this chapter, we will discuss some details of basic ECG signal, overview of Discrete wavelet transform along with Mallat algorithm for wavelet decomposition and details of various compression techniques, which can be used for the compression of the ECG signal.

3.2 The Electrocardiogram

The Electrocardiogram (ECG) signal is a time varying signal, which is a recording of the electrical activities of the heart over time produced by an Electrocardiograph and is a well-stabilized diagnostic tool for cardiac diseases. ECG signal is monitored by placing sensors at defined positions on the chest and limb extremities of the subject [2]. The electrocardiograph or ECG machine permits deduction of many electrical and mechanical defects of the heart by measuring ECGs, which are potentials measured on the body surface. Each heartbeat is caused by a section of the heart generating an electrical signal, which then conducts through specialized pathway to all parts of the heart [1]. These electrical signals also get transmitted through the chest to the skin, where they can be recorded.

An Electrocardiogram signal can be used for detection of coronary artery disease, cardiomyopathies and left ventricular hypertrophy. It can also provide information for evaluation rhythm disorders. The ECG can be obtained by recording the potential difference between various electrodes placed on the surface of the skin, at specific locations. A single normal cycle of the ECG occurs with every heart beat[2]. The electrocardiograph captures the signal through the array of electrode sensors placed at the standard locations on the skin of the human body. Modern Electrocardiographs record ECG signals by digitizing and then storing the signal in the magnetic or optical discs. The electrical waves can be measured at selectively placed electrodes (electrical contacts) on the skin. Electrodes on different sides of the heart measure the activity of different parts of the heart muscle. An ECG displays the voltage between pairs of these electrodes, and the muscle activity that they measure, from different directions, also understood as vectors. The ECG signal is composed from five waves labeled using five capital letters from the alphabet: P, Q, R, S, and T. The width of a wave on the horizontal axis represents a measure of time. The height and depth of a wave represent a measure of voltage [1].

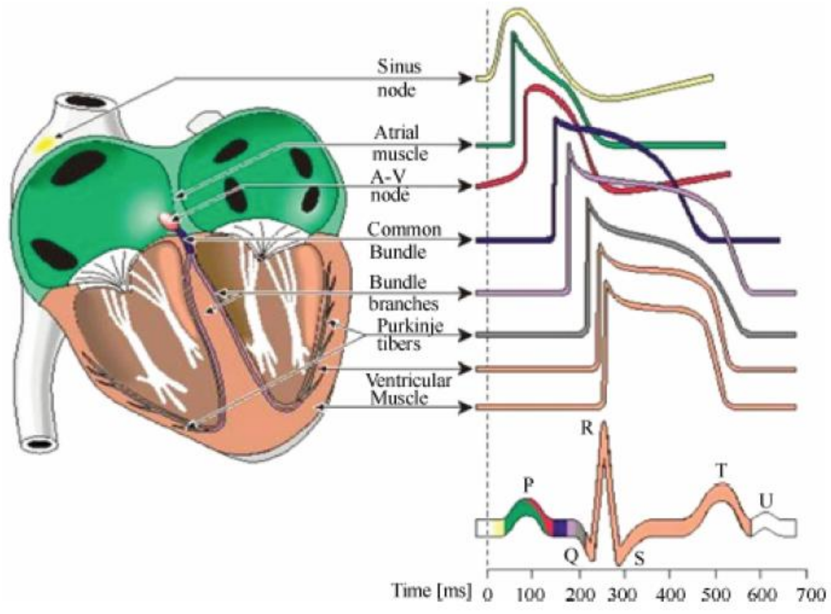

Figure 3.1: A typical representation of ECG wave

Figure 3.1 [2] shows a typical representation of an ECG wave. The ECG signal is made up of a number of segments or waves of different durations, amplitudes, and forms: 'slow', low-frequency P and T waves and short and high-frequency Q, R, and S waves, forming the QRS complex. P wave, QRS wave, and T wave, they are diagnostic critical waves. The P wave represents the atrial depolarization where the blood is squeezed from the atria to the ventricles. The QRS segment is when the ventricles depolarize and squeeze the blood from the right ventricle to the aorta. The T wave represents the period of time when the ventricles repolarize (get ready for the next heart beat). Most of the ECG signal energy is concentrated in the QRS complex, but there are diagnostically important changes in the low amplitude PQ and ST intervals, the P and T waves.

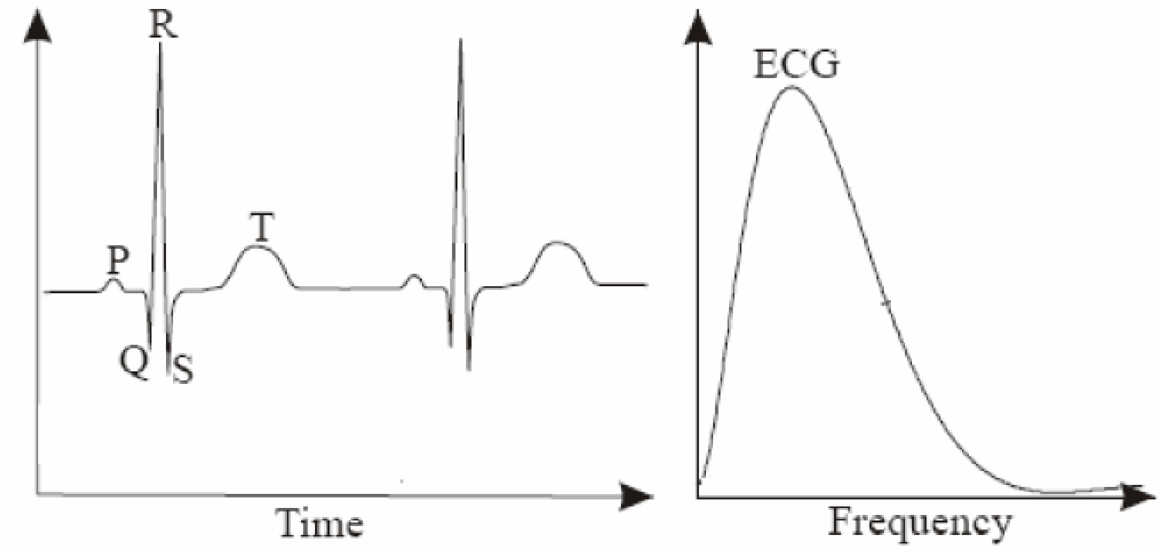

Figure 3.2: ECG Signal in time and frequency domain

Figure 3.2 [2] shows the ECG signal in time and frequency domains. Compressing the ECG signal, while preserving the original shape of the reconstructed signal and specially the amplitudes of Q, R and S peaks, without much distortions in the low amplitude ST segment, P and T waves are the main objective for the ECG compression.

The various applications of ECG are as follows:

1) In Cardiac stress test, wherein the patient walks on a treadmill while connected to an electrocardiogram machine.
2) In Screening test, which identifies coronary artery disease prior to the onset of symptoms
3) Preoperatively to rule out coronary artery disease
4) Provide information in the presence of metabolic alterations such has hypercalcemia, Hypocalcemia, hyperkalemia and hypokalemia.
5) With known heart disease, monitor progression of the disease
6) It is used in evaluation of rhythmic disorders by observing electrocardiogram
7) It is the basic cardiologic test and is widely used to diagnose patients with suspected or known heart disease.

3.3 Wavelet Theory

3.3.1 Wavelet and Wavelet Analysis

A wavelet is a waveform of effectively limited duration that has an average value of zero. They are localized waves, instead of oscillating forever, they drop to zero. They come from iteration of filters (with rescaling) [6]. If we compare wavelets with sine waves, which are the basis of fourier transform, Sinusoids do not have limited duration- they extend from minus to plus infinity, where sinusoids are smooth and predictable. Wavelets tend to be irregular and asymmetric. Wavelet analysis can be applied to one-dimensional, two-dimensional data and, in principle to higher-dimensional data [46].

Wavelets have a property called **Scaling**. Scaling a wavelet means to compress or expand the wavelet. Smaller scales, the more compressed the waveform and more detailed is the information received from it. Wavelets also have a property called **Shifting**, which is self-explanatory [46].

The main difference between fourier analysis and wavelet analysis is that fourier analysis splits up the signal (time domain) to its corresponding frequency domain, which is a drawback in case of non stationary signals like ECG signal, because these signals have sudden frequency changes with time, which when transformed using fourier transform, is not reflected in the frequency domain. Thus FT is at a disadvantage in these cases. In order to solve this problem a probable approach is the Short Time Fourier Transform (STFT) where the signal is segmented into windows, assuming that the signal stays stationary inside those windows. Thus FT forms a limiting case of the STFT, where the length of the window is infinite [46].

The STFT seems to solve the problem of non stationary signals of FT. But there is another problem that arises, that of resolution. According to the Fourier Uncertainty Principle, the time and frequency resolution are inversely proportional. A signal concentrated in time is spread out in frequency and vice versa. In STFT one cannot know exactly what frequency exists at what times, but at what time intervals what band of frequencies exist. This is due to the fact that the window of the STFT, also called the support of the function is fixed and hence the resolution is fixed. In FT, a window of infinite length realizes a perfect frequency resolution. So the tradeoff is as follows,

Narrow window → good time resolution, poor frequency resolution.

Wider window → good frequency resolution, poor time resolution.

The requirement to obtain better frequency resolution gave rise to Multi-Resolution Analysis (MRA) [46]. This refers to the fact that different frequencies are resolved at different levels. Every spectral component is not resolved equally like the STFT as shown in figure 3.3 [46].

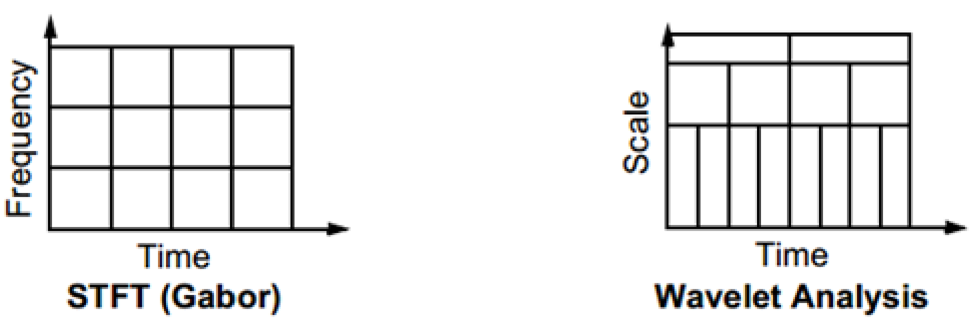

Figure 3.3: Resolution of STFT and Wavelet Transform

As shown in figure 3.3, unequal resolution can be seen in the case of the wavelet analysis. It can be seen in the case of the STFT and the MRA, the boxes have non zero area thus the particular point in time frequency cannot be known but instead a region can be identified. Additionally in the case of the MRA, the boxes at the low frequency have shorter heights thus having better frequency resolution but poor time resolution. MRA is designed to give good time resolution and poor frequency resolution at high frequencies and good frequency resolution and poor time resolution at low frequencies. This approach makes sense especially when the signal at hand has high frequency components for short durations and low frequency components for long durations. Fortunately, the signals that are encountered in practical applications are often of this type. Wavelet analysis allows the use of long time intervals where we want more precise low frequency information, and shorter regions where we want high frequency information The advantage offered by wavelets is the local analysis performed by variable sized windows.

Wavelet analysis is capable of revealing aspects of data than other signal processing techniques like aspects in trends, breakdown points, discontinuities in higher derivatives, and self-similarity, Further because it affords a different view of the data than those presented by traditional techniques, wavelet analysis can often compress or de-noise a signal without appreciable degradation.

3.3.2 Wavelet Transforms

There are two types of Wavelet Transforms:

3.3.2.1 Continuous Wavelet Transform:

Continuous wavelet transform is the sum over all the time of the signal multiplied by scaled, shifted versions of the wavelet. This process produces coefficients that are a function of scale and position. A portion of the wave is compared with the wavelet on a certain scale. This entire process is repeated over the entirety of the waveform. The wavelet is then scaled by a different factor and then the above process is repeated once again and so on.

The scales and the frequency are not unrelated. Higher scales relate to more expanded wavelets and smaller scales to compressed wavelets. The more expanded the wavelet, the longer portion of the signal is it being compared to, and hence the signal features observed are coarser as shown below:

Low scale → Compressed wavelet →Rapidly changing details → High frequency.

High scale → Expanded wavelet → Slowly changing details → Low frequency

The continuous wavelet transform can be shown as,

$$CWT(u,s) = \frac{1}{\sqrt{|s|}} \int x(t)\, \Phi\left(\frac{t-u}{s}\right) dt$$

Where ‘u’ is the shifting parameter and the ‘s’ is the scale parameter. The Φ(t) function is called the mother wavelet function, as it is used to derive different delayed and scaled versions to be used for performing the transform. This definition of the CWT shows that the wavelet analysis is a measure of similarity between the functions (wavelets) and the signal itself. Here the similarity is in the sense of similar frequency content. The calculated CWT coefficients refer to the closeness of the signal to the wavelet at the current scale.

In the case of the continuous wavelet transform, the wavelet is operated at every scale. The discrete wavelet transform instead operates on specific scales.

3.3.2.2 Discrete Wavelet Transform

Calculating wavelet coefficients at every possible scale is a fair amount of work, and it generates an awful lot of data. It turns out, rather remarkably, that if we choose scales and positions based on powers of two, so-called dyadic scales and positions, then our analysis will be much more efficient and just as accurate. We obtain just such an analysis from the discrete wavelet transform (DWT), The DWT was developed to apply the wavelet transform to the digital world. An efficient way to implement this scheme using filters was developed in 1988 by Mallat [47]. Low frequency parts of signals lend the signal quality but the high frequency part gives the signal details. For the wavelet analysis two terms are used "approximation" and "details". Approximations are the high-scale, low frequency components whereas the low-scale, high frequency parts are the details component.

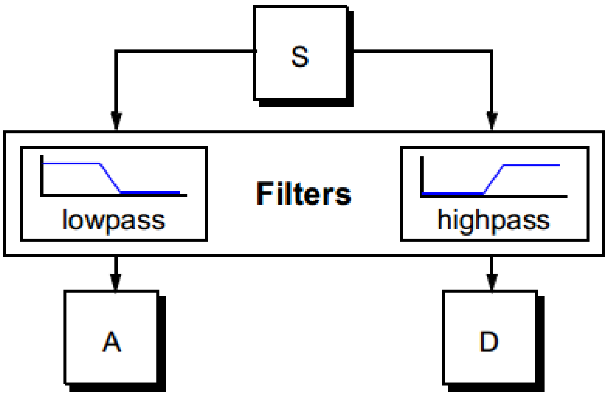

Figure 3.4: Sub-band Decomposition of Wavelet Transform.

The original signal passes through two sets of filters, as seen in figure 3.4 [46], which split the signal into two sub-bands. In a real digital signal if this operation is performed the end result is data with twice the amount.

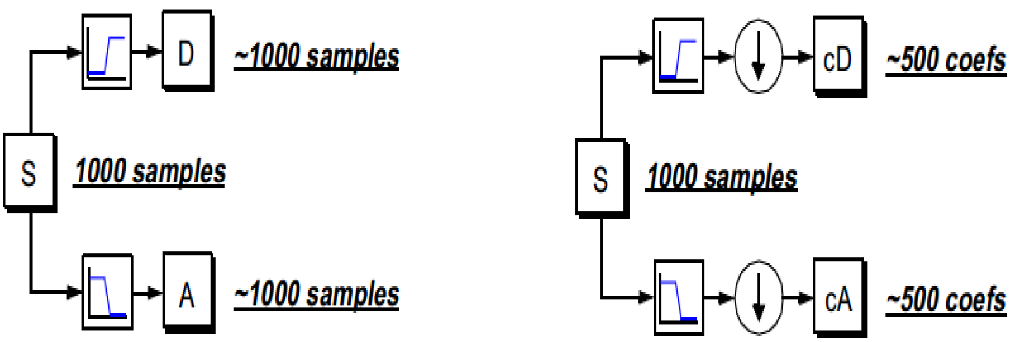

Figure 3.5: Down-sampling operation in wavelet transform.

Figure 3.5 [46] shows the Down-Sampling operation in wavelet transform. The reduction of samples is done by a down sampling operation where the data samples from alternate sample points are removed. The complementary filter gives rise to Quadrature Mirror Filters (QMF). The two filters are related by the equation,

$$g[L-1-n] = (-1)^n h[n] \quad \text{.....................................(3.1)}$$

Where g(n) is the highpass and h(n) is the lowpass filter, L is the number of sample points.

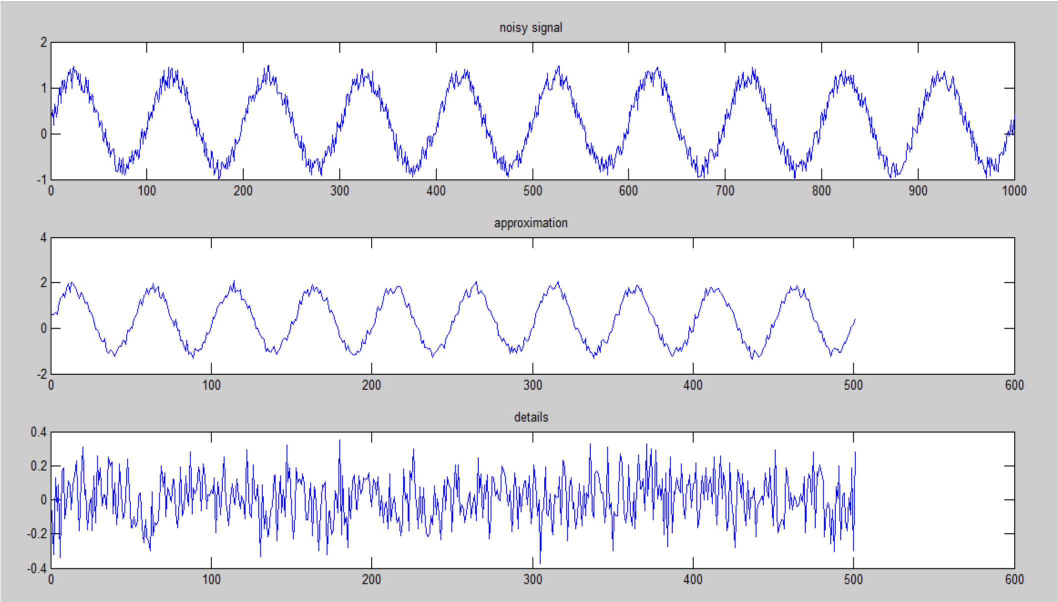

Figure 3.6: DWT of a noisy sinusoid

MATLAB CODE:

```
S = sin(20.*linspace(0,pi,1000)) + 0.5.*rand(1,1000);

[cA,cD] = dwt(s,'db2');

plot(cA)

plot(s)

plot(cD)
```

It can be seen in figure 3.6 [48] that the approximation has the structure of the sinusoid, whereas the details have the noise information or the high frequency information. It can be seen that by this approach the noisy signal can be de-noised without losing appreciable quality. This decomposition can be carried out to multiple levels depending on the requirements of the application. Theoretically the decomposition can be carried out infinite number of times but is usually selected on the basis of certain criteria like entropy. Successive decomposition gives resulting Wavelet Decomposition Tree.

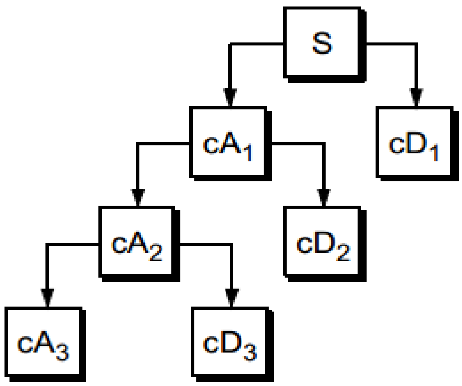

Figure 3.7(a)

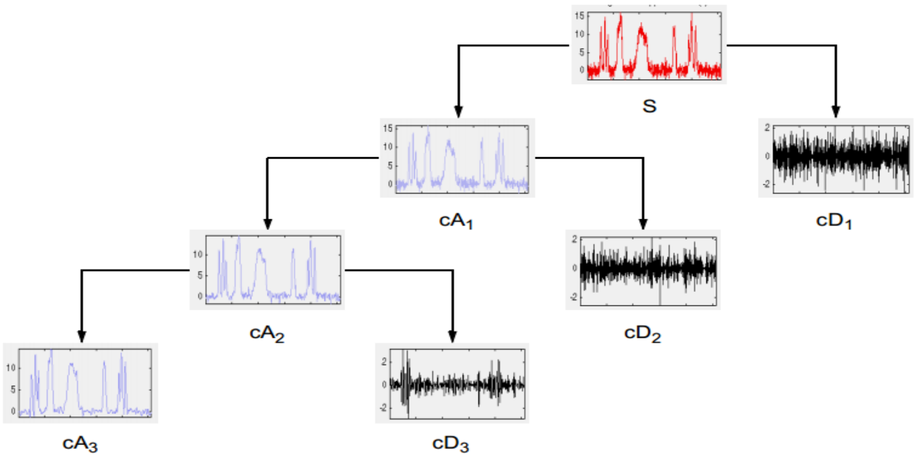

Figure 3.7(b)

Figure 3.7 (a) and (b): Wavelet Decomposition Tree for a noisy signal S.

Figure 3.7(a) and 3.7(b) shows the wavelet decomposition tree for a noisy signal with three level wavelet decomposition.

Down-sampling of signal components causes aliasing. The filters required for reconstruction are chosen carefully to avoid this.

Filter banks are used to approximate the behavior of the continuous wavelet transform [48]. The signal is decomposed with a high pass filter and a low pass filter. The coefficients of these filters are calculated using mathematical analysis and are available to us,

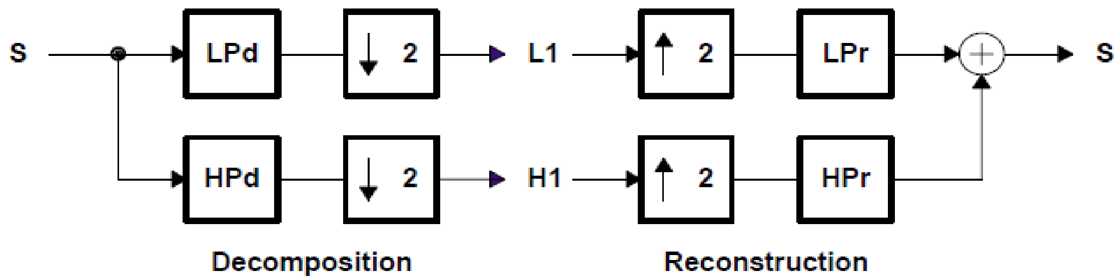

Figure 3.8: Discrete Wavelet Transform

Figure 3.8 [46] shows the one level decomposition and reconstruction of signal S using filter banks and down-sampling and up-sampling respectively.

Where LPd: Low Pass Decomposition Filter

HPd: High Pass Decomposition Filter

LPr: Low Pass Reconstruction Filter

HPr: High Pass Reconstruction Filter

The coefficients for the filters HPd, LPd and HPr can be computed from the h(n) coefficients as shown below [7]:

1) High-Pass decomposition filter (HPd) coefficients are given by,

$$\bar{g}(n) = (-1)^n \, h[L-n] \quad \ldots(3.2)$$

2) Low-Pass reconstruction filter (LPr) coefficients are given by,

$$\bar{h}(n) = h[L-n] \quad \ldots(3.3)$$

3) High Pass reconstruction Filter (HPr) coefficients are given by,

$$\bar{g}(n) = g[L-n] \quad \ldots(3.4)$$

Where L represents the length of the filter.

If the length of the filter is high, then it provides smoother, smaller and intermediate results. Thus if intermediate results are required, then it is less likely that we lose information due to necessary threshold or saturation and also use of longer filters will involve more processing.

Filter banks are used to decompose the signal into high and low frequency components. The low frequency components, called the approximation signal, contains most of the frequency of the signal, whereas high frequency signal contains the details of the signal. Wavelet decomposition can be implemented using the two-channel filter bank. The main idea is that the perfect reconstruction filter banks implement the series expansions of discrete-time signals.

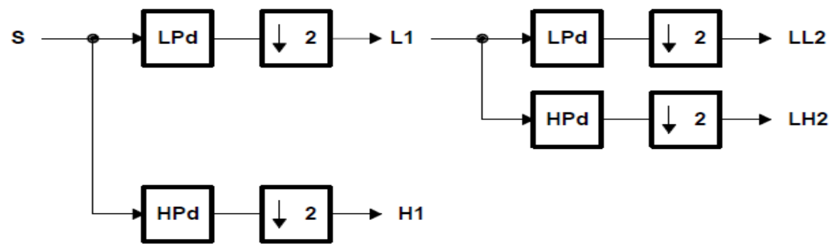

Figure 3.9(a)

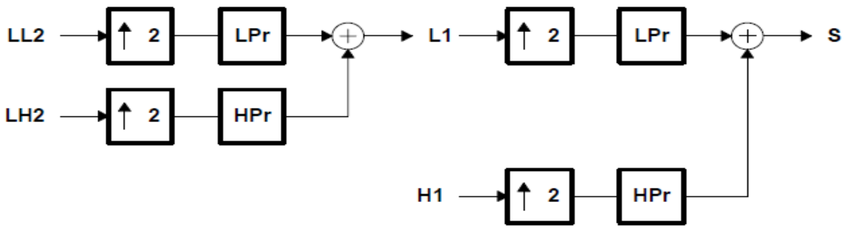

Figure 3.9(b)

Figure 3.9 (a) and (b) : A two-level wavelet decomposition and reconstruction.

If the input and the reconstruction are identical, then it is called the perfect reconstruction. There are two types of decomposition structures namely pyramid and wavelet packet. Pyramid structure only decomposes approximation i.e. low-frequency component, while wavelet packet structure decomposes both the approximation (low-frequency component) and details (high-frequency component).

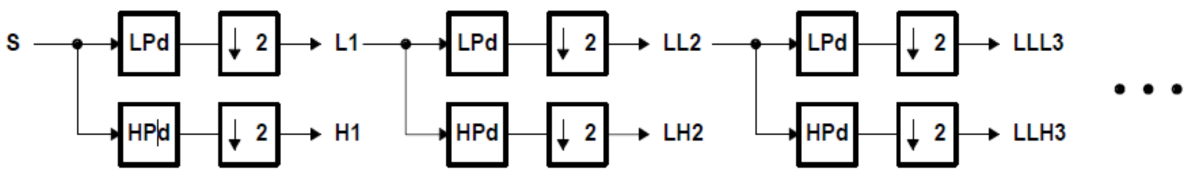

Figure 3.10: Pyramid Packet Decomposition

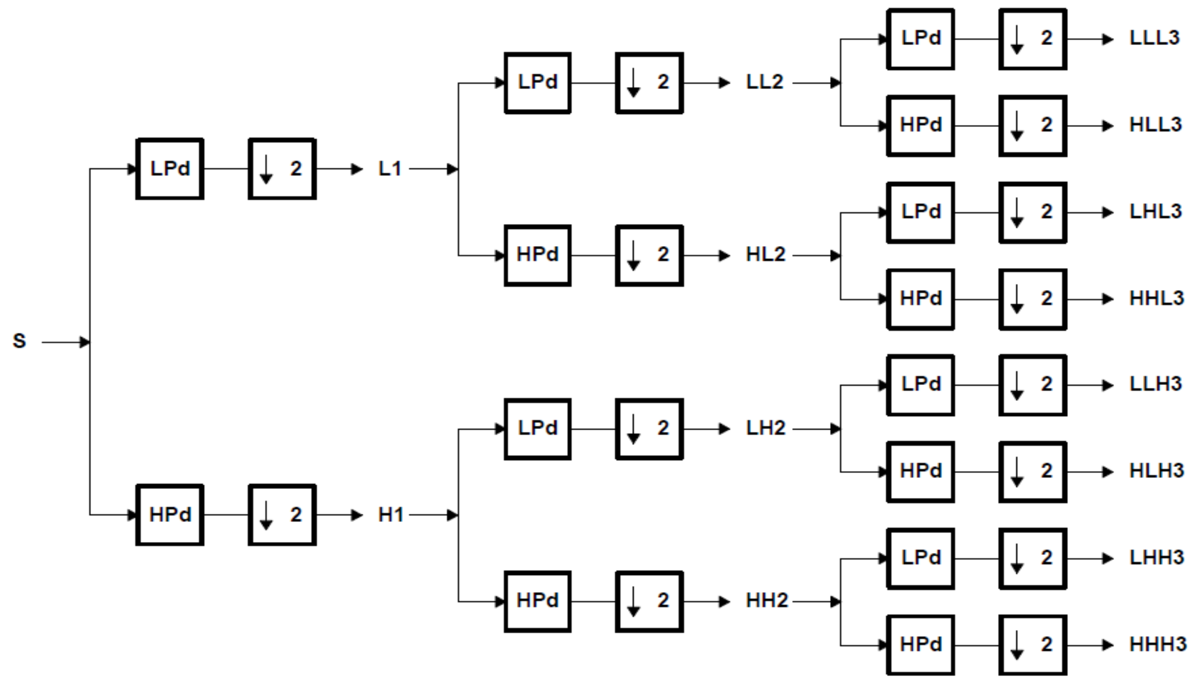

Figure 3.11: Wavelet Packet Decomposition

Figure 3.10 and 3.11 [46] shows the pyramid packet decomposition and wavelet packet decomposition respectively. In this work we have used pyramid packet decomposition method for the ECG compression.

3.3.3 Different Wavelet Families

There are a variety of wavelet families [50], such as: Daubechies, Symmlet, Meyer, Morlet, Haar or Coiflet, etc. The qualities of elements of these families vary according to several criteria: the length of the support of their mother wavelet, the number of vanishing moments, the symmetry or the regularity. Another two criteria are of high importance are the existence of corresponding scaling function and the orthogonality and non-orthogonality of the analysis. Within each wavelet family there are wavelet subclasses differentiated by the number of coefficients and by the level of iteration.

There are different wavelet families, which are required for different purposes. These families of wavelets are :

1) **Haar**: Simplest of wavelet families and resembles a step function. It comes from the family of orthogonal wavelets. It is not adapted for approximating smooth signals, because it has only one vanishing moment. Haar wavelet transform has the advantage that it is conceptually simple, fast and also memory efficient and it is a good choice to detect the time localized information.

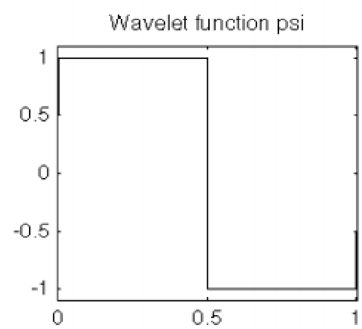

Figure 3.12(a): A simple Haar wavelet.

2) **Daubechies**: It represents a family of orthogonal wavelets, which is characterized by the maximum number of vanishing moments for some given support's length. Corresponding to each mother wavelet from this family, there is a scaling function, which is also called the father wavelet, which generates an orthogonal MRA. The elements of this family, which are mostly used in practice are db2-db20, where the index refers to the number of vanishing moments, means db1(Haar wavelet) has only one vanishing moment, db2 will have two vanishing moments and so on. The Daubechies mother wavelets are not symmetrical.

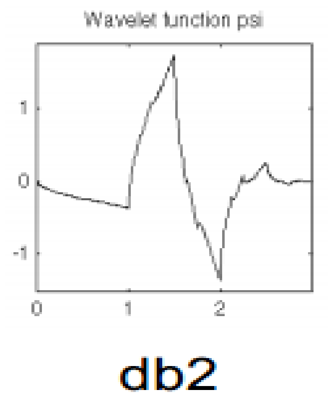

Figure 3.12(b): Daubechies2 (db2) wavelet

3) **Bi-orthogonal**: This type of wavelet family display linear phase properties required for signal and image reconstruction. Two different wavelets, one for decomposition and other for reconstruction, interesting properties can be derived.

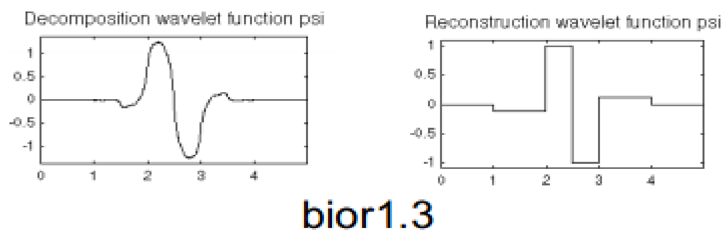

Figure 3.12(c): Bi-orthogonal wavelet.

4) **Coiflet:** This type of wavelet family's mother wavelet is more symmetrical than the Daubechies mother wavelet.

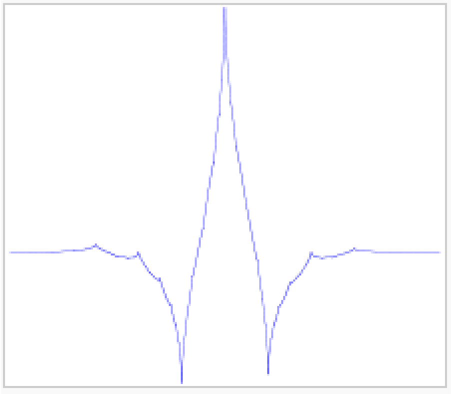

Figure 3.12(d): Coiflet wavelet with two vanishing moments

5) **Mexican hat**: Derived from a function proportional to the second derivative of the Gaussian probability density function.

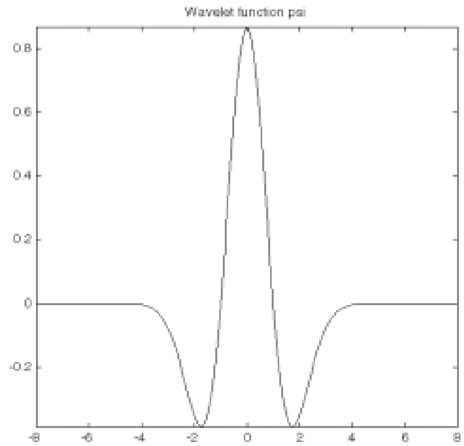

Figure 3.12(e): Mexican hat wavelet

Other wavelet families like the coiflets, symlets, morlets etc. exist for other purposes with different properties. Some of the application regarding the wavelets include detecting discontinuities and breakdown points, detecting long term evolution, detecting self-similarity, identifying pure frequencies, suppressing signals, de-noising signals, compressing signals. Thus it can be seen that it is possible to perform the discrete wavelet transform on multitude of data signals to extract different sources of information and perform certain key operations on them subsequently.

3.4 ECG Signal Compression with Various Compression Techniques

Compression of signals is based on removing the redundancy between neighboring samples and between the adjacent cycles [51]. In data compression, it is required to represent data by as small as possible number of coefficients within an acceptable loss of visual quality of the signal. Compression techniques can be classified into two main categories: lossless and lossy compression. Lossless compression methods are fully reproducible and their decompression should result in the exact reconstruction of the original signal. Lossy compression techniques are those, which may or may not be fully reproducible. Although the result of decompression may not produce an exact duplication of the original signal, the differences between the original and reconstructed signal may be so small. Since the compression ratio, which can be obtained by the use of lossy compression technique can significantly exceed the compression ratio obtainable from lossless compression technique, the primary tradeoff is between reproducibility versus the requirements for storage space and transmission rate of the signal. Compression methods can be classified into three functional categories [8,52]:

3.4.1 Direct Signal Compression Methods:

Direct Methods involve the compression performed directly on the ECG signal. These are also known as time-domain techniques, where the samples of the ECG signal are directly handled to provide the compression. Coding is done by time-domain methods, which is based on the idea of extracting a subset of significant signal samples to represent the signal. To get a high performance time domain compression algorithm, efforts should be make in the designing of Intelligent Sample Selection Criteria. The original signal is reconstructed by an inverse process, more often straight line are drawn between the extracted samples. Examples of methods, which belong to this group are Turning Point (TP) method, Amplitude Zone Time Epoch Coding (AZTEC) method, and Coordinate Reduction Time Encoding System (CORTES) method [51,53]. For successful algorithm, the key is to have a good rule to determine the most significant samples. The more recent cardinally constrained shortest path Technique also fits into this category of compression. Decoding in this method is based on interpolating the subset of samples. The traditional ECG time domain Compression algorithms are based on heuristics in the sample selection process, which generally make them fast.

3.4.2 Transformed ECG Compression Methods

Transform domain methods, as their name implies, operate by first transforming the ECG signal into another domain and then compression is done in that domain. These methods mainly utilize the spectral and energy distributions of the ECG signal by means of some transform, and properly encoding the transformed output [8]. Signal reconstruction is achieved by applying the inverse transformation process. This category includes the traditional transform coding techniques, which are applied to the ECG signals such as the Karhunen–Loeve transform , Fourier transform [46], Cosine transform , sub-band techniques , vector quantization (VQ) [15] and most recently used the wavelet transform (WT) [54].

Wavelet technique is obviously the first choice for ECG signal compression because of its localized and non-stationary property and also the well-proven ability of wavelets to see through signals at different resolutions. Wavelets are mathematical functions, which cut up data into different scale-shift components [2]. The wavelet decomposition splits the analyzing signal into approximation and detail coefficients, using finite impulse response digital filter banks. The main task when wavelet analysis (decomposition and reconstruction) is used, is to find a good analyzing function (mother wavelet) to perform an optimal decomposition. Wavelet-based ECG compression algorithm have been proved to give the good compression results. The ability of DWT to separate out the signal components has given rise to a number of wavelet-based compression techniques which provide much better compression results than those based on traditional Fourier methods. The discrete wavelet transform has interesting mathematical properties and it suitably fits in with the standard signal filtering and encoding methodologies. It produces few coefficients than the original signal and the user does not have to worry about losing energy during the transform process or its inverse. The DWT is faster and maps quickly to the sub-band coding of signals, the Continuous Wavelet Transform (CWT) allows the user to analyze the signal at various scales and translations according to the problem [2].

3.4.3 Optimization Methods for ECG Compression

In this type of technique, a preprocessor is employed to the signal to extract some features that are later used to reconstruct the signal. The main goal of these types of techniques is to minimize the reconstruction error with a given bound that number of samples to be extracted, In this method ECG signal is compressed by extracting the signal samples that, after interpolation, will best represent the original signal with an upper bound on their number. After the samples are extracted, they are Huffman encoded. This leads to the best possible representation in terms of the number of extracted signal samples, but it is not necessary that it is in terms of the bits used to encode these samples. The bit rate is taken into consideration at the optimization stage. Examples of the methods belonging to these groups are linear prediction methods [55] and neural networks [16] methods. Recently, there has been some activity using these methods, hybrid time–frequency domain methods are integrated; namely linear prediction and wavelet transforms.

The majority of the above mentioned methods do not permit the exact reconstruction of the original ECG signals. To preserve the clinical diagnostic features of the reconstructed ECG signal, both the wavelet filter parameter and the threshold used in all subbands should be carefully selected. Thus, the main aim of the ECG signal compression is to get the best technique that can achieve maximum data volume reduction and also preserving the significant signal morphology features on reconstruction, which can be achieved by minimizing the bit rate and also the distortion of the reconstructed ECG signal using the parameterization of wavelet filters and selecting the optimum threshold level of the wavelet coefficients in different sub-bands [2].

3.5 Different Coding Techniques for ECG Compression

The DWT transforms the ECG signals into sub-bands that needs to be encoded to get the required compression result. The selection of suitable coding technique is very important to get the required compression performance with minimized reconstruction error. Several coding methods such as Vector Quantization (VQ) [15], Set Partitioning In Hierarchical Tree (SPIHT) coding [15,28] and Energy Packing Efficiency (EPE) [56] etc. We can summarize these techniques as follows:

3.5.1 Vector Quantization (VQ):

The Vector Quantization (VQ) [15] utilizes a codebook to match the wavelet coefficient vector by an index pointing to the closest code-vector in that codebook. It can achieve the excellent performance in the ECG compression.

This technique has the disadvantage that code-vector searching and codebook replenishment and storage increases the complexity of implementation in VLSI or DSP board.

3.5.2 Set Partitioning In Hierarchical Tree (SPIHT) coding:

In this coding technique, the wavelet coefficients are linked as a patient-offspring Orientation tree, which is ordered from low to high frequency sub-bands [15,28]. An initial Threshold is set as $2^{[\log_2 \max(|C_i|)],}$, where $|C_i|$ represent the magnitude of the wavelet coefficient. In SPIHT algorithm, the wavelet coefficients are manipulated based on this threshold. If the coefficient is found to be significant with respect to the above threshold, It is refined by subtracting the refinement value. After this a sequence of coded bits is generated to record the manipulation process, then a new partition pass is performed on the refined hierarchical tree based on the new threshold. This partition pass is continued till a desired quality with the specified bit rate is achieved. In addition to this, SPIHT has the advantage of embedded zero-tree wavelet, which uses fewer bits to remove the insignificant sub-tree.

SPIHT coding scheme has the disadvantage that it requires a repetitive manipulation of the wavelet coefficients and also it requires a complex sorting scheme to move the wavelet coefficients among the lists of the significant points, insignificant points and insignificant sets. The complexity of implementation of this coding technique is much more higher.

3.5.3 Energy Packing Efficiency (EPE):

EPE [56] uses the individual threshold to search the significant coefficient in each sub-band. The coefficients in each sub-band are first stored descending in magnitude to find the appropriate threshold. The thresholded coefficients are then subsequently squared and accumulated. The sub-band threshold is calculated based on the cutoff value, where the accumulated energy reaches the desired percentage of the total energy.

This method also has the disadvantage that, there are several heavy computations such as squaring, sorting and energy accumulation are needed for this method, which also increases the complexity of the coding architecture.

There are also several other coding techniques, which can be used, but all these techniques are complex to implement on FPGAs, Microcontrollers and Digital Signal Processors (DSPs) and also they require high computational cost, which make them unsuitable for the wearable battery driven health monitoring system.

In this work, we have used, a coding technique Bit-Field preserving combined with the Running Length Encoding.

3.5.4 Bit-Field-Preserving (BFP) and Running Length Encoding (RLE):

This is a very efficient coding technique, which overcome the problems associated with the above coding techniques. The BFP-RLE [57,58] is performed in two stages. In the first stage bit-depth of each sub-band, obtained after Discrete-Wavelet-Transform (DWT), is defined. In the second stage, coding of the computed wavelet coefficients is done based on the computed sub-band bit-depths and the preset preserved bit-lengths. This method introduces an effective coding technique to overcome the case, when the compressed packet exceeds the payload size by splitting the ECG data into smaller blocks and then recompressing them.

In this work, we have used this DWT-BFP-RLE technique for compression of ECG signal on C6713 Digital Signal Processor , which provide good compression result and also very easy to implement. Running Length Encoding is a coding technique, which is very easy to implement and provide an efficient technique for the compression of ECG signal. The DWT-BFP-RLE technique will be discussed in detail in next chapter.

3.6 Compression Algorithm Performance Measures:

Compression algorithm all aim to remove the redundancy within data, thereby discarding Irrelevant information. In case of ECG compression, data that does not contain any diagnostic information can be removed without any loss to the physician. The criteria for testing the performance [59] of the compression algorithms consists of three components: Compression Measure, Reconstruction Error and the Computational Complexity. The Compression measure and the reconstruction error depend on each other and determine the rate distortion function of the algorithm.

3.6.1 Compression Ratio (CR): The Compression ratio is defined as the ratio of the number of bits representing the original signal to the number of bits required for representing the compressed signal. The CR calculates the ratio between the number of bits in the original signal block (b_{or}) and the number of bits in the compressed packet (b_{comp}).

$$CR=(b_{or} / b_{comp})\text{……………………………………………………}(3.5)$$

3.6.2 Percentage Root Mean Square Difference (PRD): The PRD is used as an error estimate. It is defined as the percentage root mean square difference between original signal x_{or} and the reconstructed signal x_{re}.

$$PRD=\sqrt{\frac{\sum(x_{or}-x_{re})^2}{\sum(x_{or})^2}}\times 100\%\text{………………..………………..}(3.6)$$

CR and PRD are the most important compression measures in all the literature given in the above literature survey. Thus a new measure called the Quality Score was suggested by Fira and Goras [16]. If the quality score is high then it represent the good compression performance. Thus quality score is given by,

$$QS = CR / PRD \qquad (3.7)$$

These parameters can just evaluate the compression scheme's efficiency and signal quality, but they do not represent the clinical quality of the reconstructed signal, for this visual inspection by the cardiologist is must in some cases.

3.7 Summary

In this chapter, the basics of ECG signal is discussed. Details of Discrete Wavelet Transform has also been discussed. Along with this various techniques available for compression of ECG signal has also been presented. This chapter gives a brief idea of, how wavelet transform is better than other transforms. Advantages of BFP-RLE scheme over other schemes have also been presented.

Chapter 4
Implementation of the Compressed ECG Signal

4.1 Introduction

Digital Signal Processing has the flexibility to perform complex mathematical operations in the digital domain. They can also provide stability, re-configurability in implementation of many DSP applications, Because of these reasons today digital signal processing is used in many applications to maximize the use of digital components in high performance systems. ECG compression is such an application, which can be implemented on a Digital Signal Processor to reduce the payload of the real time ECG transmission as well as to reduce the amount of data storage in long term ECG recording. Thus the mathematical operations that can't be implemented in analog domain can be easily implemented in digital domain using Digital Signal Processing.

Section 4.2 explains, the generation of ECG signals from MIT-BIH Arrhythmia database and also the wavelet decomposition of these ECG signals is presented. Section 4.3 explains, how these decomposed signals can be encoded using Bit-Field Preserving and Running Length Encoding Techniques, The program flow chart of these techniques are also presented. This section also explains, how this compressed ECG can be packetized for transmission. Section 4.4 explains the decompression or reconstruction of the compressed ECG signal. Section 4.5 explains the various Performance parameters that are used to determine the accuracy of the result. Finally Section 4.6 give some experimental results for different ECG signals.

4.2 Wavelet Decomposition of the ECG Signal

4.2.1 Generation of ECG signal

The ECG signal data are taken from the MIT-BIH Arrhythmia database [62] available at PHYSIONET, where several records 100, 101, 102 etc. of the ECG signals are available. The ECG signals at MIT-BIH database are sampled at 360 Hz with a resolution of 11-bit/sample. In this database 10 seconds, 1 minute, 10 minutes, 1 hour, 10 hour and to end lengths of different ECG records are available. The ECG signals .mat file is taken from this database to generate the header file of ECG coefficients of each record using matlab.

The ECG signal is preprocessed by subtracting 1024, so that the signal value fall between -1024 to 1023. We take first 10 minutes of each record and divide this whole frame into several frames, each of which contains 1024 samples. The matlab code [63] for generating the coefficient header file for ECG record say 100 is as follows:

```
%Input file = 100m.mat
load ('100m.mat')
val = (val - 1024);
ECGsignal = val(1,1:1024); % Utilizing the first 1024 values

ECGHeaderFid = fopen('ecg100.h','wt');
fprintf(ECGHeaderFid, '#define N %d \n float h [N] = {',length(ECGsignal));

ECGSigLen = length(ECGsignal);

%writing the entire array as a header filer
for i = 1:ECGSigLen

%write the value as a single precision float
fprintf(ECGHeaderFid, '%f',ECGsignal(i));

%write a comma except for the last value
if i ~= ECGSigLen
fprintf(ECGHeaderFid, ',');
end
end
fprintf(ECGHeaderFid, '};');

%close the file descriptor
fclose(ECGHeaderFid);
```

This matlab code will generate a header file ecg100.h, which will have first 1024 coefficients of ECG signal record 100. The plot of this ECG signal using Code Composer Studios graph view is as follows:

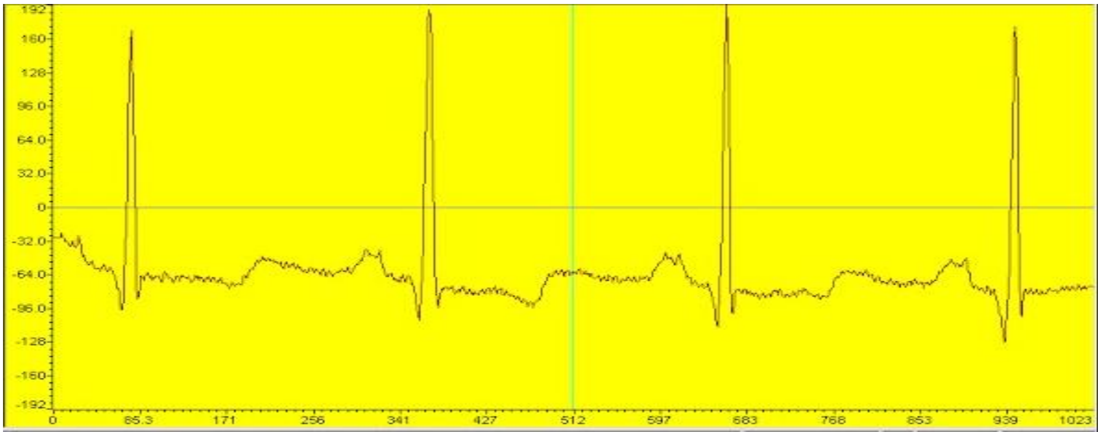

Figure 4.1: ECG Signal for mita record 100 with 1024 coefficients.

Similarily we can generate the coefficients of different ECG signal mita records like 100, 107, 117, 119 etc. Then this ECG signals can be processed further for applying DWT and coding algorithms.

4.2.2 Discrete Wavelet Transform (Mallat Algorithm)

Discrete Wavelet Transform (DWT) transform the ECG signal into the time-frequency sub-bands, which make it easy to decompose the signal into different waveforms. This wavelet based technique is best suited with the standard signal filtering methods and encoding schemes to produce the good compression results [66]. Using DWT, the ECG signal can be decomposed into J decomposition levels using low pass g(n) and high pass h(n) FIR filter banks and then they are down-sampled by a factor of 2, The decomposed signal in each level are divided into the two frequency components a low frequency signal (a_n) called the approximation signal and a high frequency signal (d_n) called the detail signal. The low frequency signal a_n is again decomposed into two signals in the next level and so on upto d_J and a_J.

These Low Pass and High Pass filter banks are constructed from any one of the wavelet basis fuction such as Daubechies, Biorthogonal, Coiflet, Symlet, Morlet and Maxican Hat. We can select any one of the Wavelet Transform function based on our application.

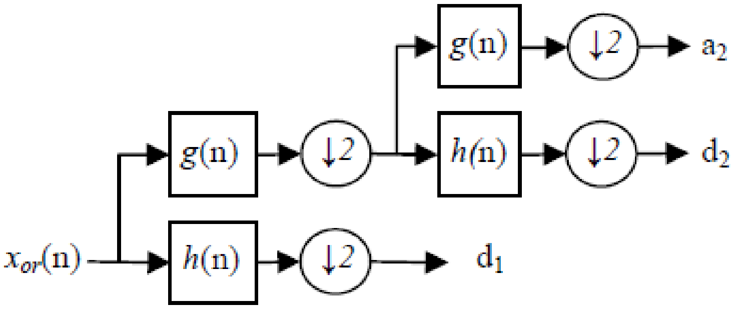

Figure 4.2: Two level DWT decomposition

Figure 4.2 [65] shows the two levels of DWT decomposition. The decomposed signal can be again reconstructed into the original signal using the reconstruction filter banks, which is inverse of the decomposition filter banks.

Since the ECG signal is a non-stationary signal, which has varying signal amplitudes with respect to time, DWT showed its powerfulness to decompose different ECG waveforms. It is found that the decomposition levels between 4 and 6 gives the best compression result [66].

Daubechies wavelets have the similarity with the ECG signal that they have similar morphology to the QRS complex of the ECG signal and it has the ability to pick up the minute details of the ECG signal.

In this work, the Daubechies wavelets with four decomposition levels, means J=4 are adopted. This Db4 filter banks with 8 coefficients are enough to give the best and fast compression performance.

In four level (J=4) wavelet decomposition of the ECG signal, at the output side we will get d_1, d_2, d_3, d_4 and a_4, which are called the sub-bands. The coefficients of these sub-bands are extracted, then they are stored in the memory and they are further processed for BFP and RLE, first this operation is done on the first 1024 coefficients of the ECG signal. Then all the frames are processed to get the required compression results [64].

For decomposing the ECG signal using mallat algorithm [48], first we require the low pass h(n) and high pass g(n) filter bank coefficients, which can be calculated using some mathematical operations or they are readily available in MATLAB [68]. The function used for getting these coefficients in MATLAB is,

Load db4, which will load the low pass filter bank h(n) coefficients for daubechies 4 wavelet, similarly we can get the coefficients for any other wavelet family using matlab.

Filter coefficients	Daubechies 2	Daubechies 3	Daubechies 4	Daubechies 5	Daubechies 6
h_0	0.3415	0.2352	0.1629	0.1132	0.0789
h_1	0.5915	0.5706	0.5055	0.4270	0.3498
h_2	0.1585	0.3252	0.4461	0.5122	0.5311
h_3	-0.0915	-0.0955	-0.0198	0.0979	0.2229
h_4		-0.0604	-0.1323	-0.1713	-0.1600
h_5		0.0249	0.0218	-0.0228	-0.0918
h_6			0.0233	0.0549	0.0685
h_7			-0.0075	-0.0044	0.0195
h_8				-0.0089	-0.0223
h_9				0.0024	0
h_0					0.0034
h_{11}					0

Table 4.1: Filter coefficients for Daubechies 2,3,4,5,6 wavelet family.

Table 4.1 [68] shows the low low pass filter coefficients of different Daubechies wavelet families. In this work we have used Daubechies 4 with 8 coefficients to get the better compression results. The high pass coefficients for db4 are calculated as follows:

$$g_0= -h_{11},\ g_1=h_{10},\ g_2=-h_9,\ g_3=h_8,\ \ldots..,\ g_{11}=h_0.$$

When we use the filter bank algorithm [46] to do wavelet transform, the decomposed signal's length is given by:

$$(Length_{signal} + Length_{filter} -1)/2 \quad \ldots\ldots\ldots(4.1)$$

And after reconstruction, the reconstructed signal length is given by,

$$Length_{signal} + 2*Length_{filter} - 2 \quad \ldots\ldots\ldots(4.2)$$

So head and tail needs to be cut-off, or we can use the circular buffer to make the reconstructed signal exactly the same as the original signal. For doing this, the following are used:

Suppose for an example the signal has 10 samples $\{S1, S2, S3, S9, \ldots, S10\}$ and a pair of filters have six elements each as given below:

$$\{\bar{g}_0, \bar{g}_1, \bar{g}_2, \ldots, \bar{g}_5\} \text{ and } \{\bar{h}_0, \bar{h}_1, \bar{h}_2, \ldots, \bar{h}5\}$$

The reconstruction filters coefficients are,

$$\{g_0,g_1,g_2,g_3,g_4,g_5\} \text{ and } \{h_0,h_1,h_2,h_3,h_4,h_5\}$$

The decomposed signal should be,

$$\{\bar{a}_1,\bar{a}_2,\bar{a}_3,\bar{a}_4,\bar{a}_5\} \text{ and } \{\bar{d}_1,\bar{d}_2,\bar{d}_3,\bar{d}_4,\bar{d}_5\}$$

To do the decomposition we do the convolution as follows:

$$a_1= S_1\bar{g}_5 + S_2\bar{g}_4 + S_3\bar{g}_3 + S_4\bar{g}_2 + S_5\bar{g}_1 + S_6\bar{g}_0$$

$$a_2= S_3\bar{g}_5 + S_4\bar{g}_4 + S_5\bar{g}_3 + S_6\bar{g}_2 + S_7\bar{g}_1 + S_8\bar{g}_0$$

After convolution down sampling is required, which is avoided by picking every other output as our output. So this convolution is shifted by 2 in place of 1, and thus the down-sampling operation is unnecessary. This algorithm can be written in the matrix form as shown in figure 4.3 [46].

$$
\begin{bmatrix}
\bar{g}_5 & \bar{g}_4 & \bar{g}_3 & \bar{g}_2 & \bar{g}_1 & \bar{g}_0 & & & & \\
\bar{h}_5 & \bar{h}_4 & \bar{h}_3 & \bar{h}_2 & \bar{h}_1 & \bar{h}_0 & & & & \\
 & & \bar{g}_5 & \bar{g}_4 & \bar{g}_3 & \bar{g}_2 & \bar{g}_1 & \bar{g}_0 & & \\
 & & \bar{h}_5 & \bar{h}_4 & \bar{h}_3 & \bar{h}_2 & \bar{h}_1 & \bar{h}_0 & & \\
 & & & & \bar{g}_5 & \bar{g}_4 & \bar{g}_3 & \bar{g}_2 & \bar{g}_1 & \bar{g}_0 \\
 & & & & \bar{h}_5 & \bar{h}_4 & \bar{h}_3 & \bar{h}_2 & \bar{h}_1 & \bar{h}_0 \\
\bar{g}_1 & \bar{g}_0 & & & & & \bar{g}_5 & \bar{g}_4 & \bar{g}_3 & \bar{g}_2 \\
\bar{h}_1 & \bar{h}_0 & & & & & \bar{h}_5 & \bar{h}_4 & \bar{h}_3 & \bar{h}_2 \\
\bar{g}_3 & \bar{g}_2 & \bar{g}_1 & \bar{g}_0 & & & & & \bar{g}_5 & \bar{g}_4 \\
\bar{h}_3 & \bar{h}_2 & \bar{h}_1 & \bar{h}_0 & & & & & \bar{h}_5 & \bar{h}_4
\end{bmatrix}
\begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \\ s_8 \\ s_9 \\ s_{10} \end{bmatrix}
=
\begin{bmatrix} a_1 \\ d_1 \\ a_2 \\ d_2 \\ a_3 \\ d_3 \\ a_4 \\ d_4 \\ a_5 \\ d_5 \end{bmatrix}
$$

Figure 4.3: Filter bank algorithm in the matrix form.

These calculations can be easily implemented on C6713 DSP using the circular buffer management.

In the same way we decompose the ECG signal using db4 wavelet four times to get the detail coefficients of d_1, d_2, d_3, d_4 and the approximation coefficients of a_4.

The output for the four level wavelet decomposition using Code Composer Studio is as follows:

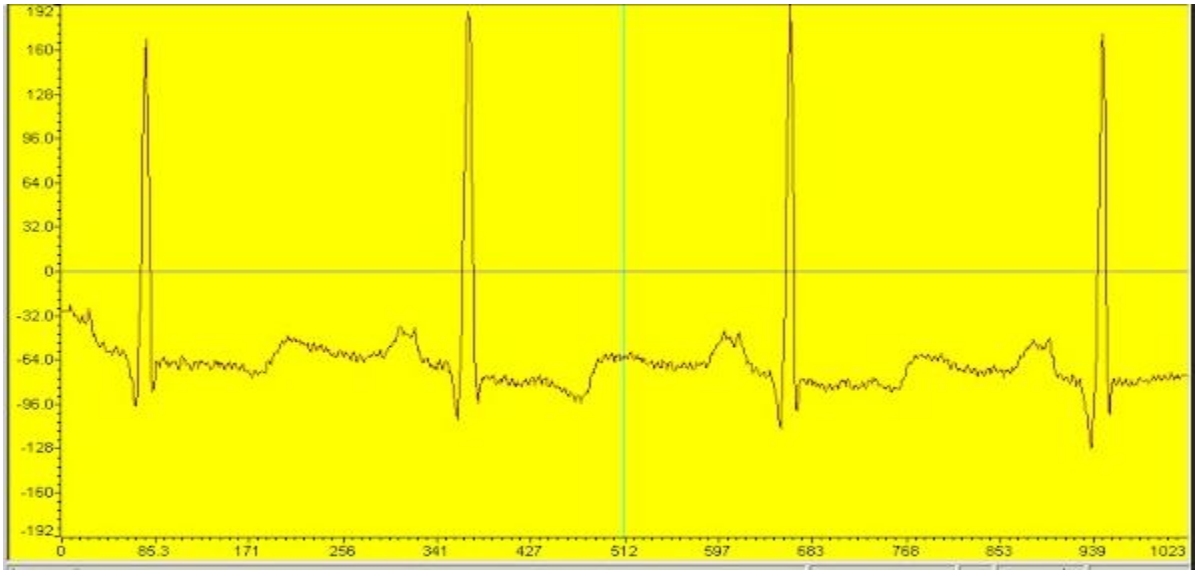

Figure 4.4(a): ECG mita record 100 with first 1024 coefficients.

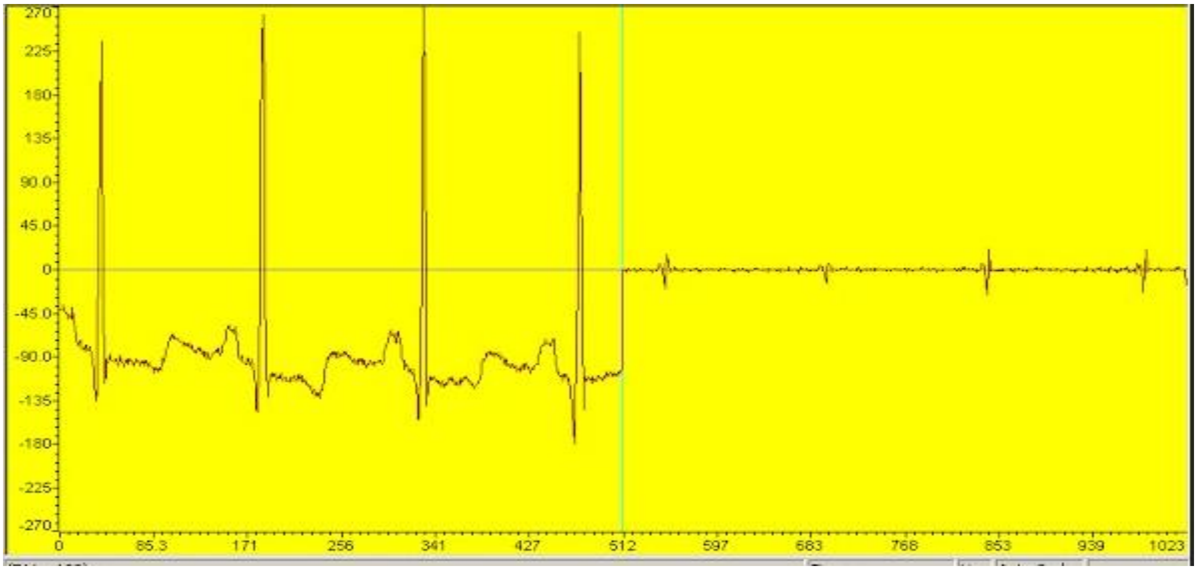

Figure 4.4(b): ECG mita record 100 with one level wavelet decomposition with approximation a_1 to the left and detail d_1 to the right.

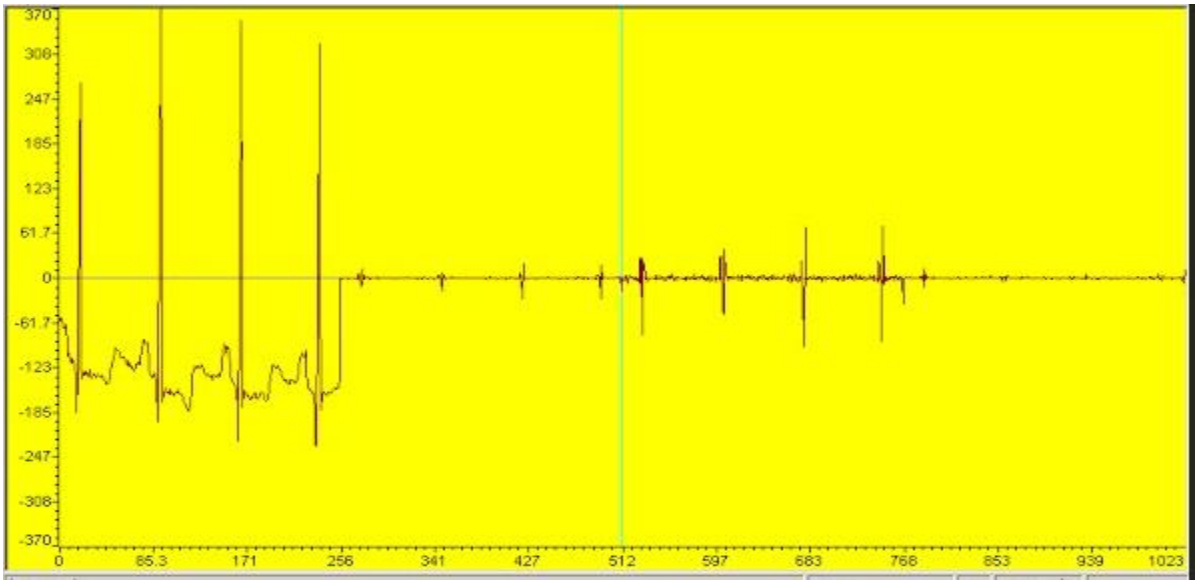

Figure 4.4(c): ECG mita record 100 with two level decomposition producing a_2 and d_2.

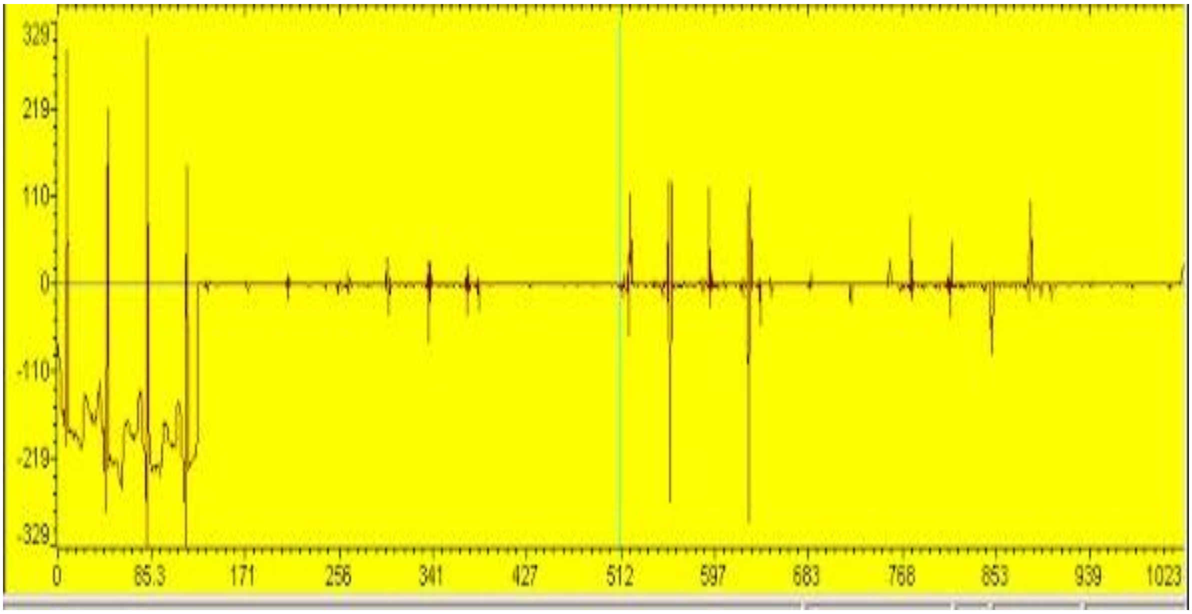

Figure 4.4(d): ECG mita record 100with third level decomposition producing a_3 and d_3.

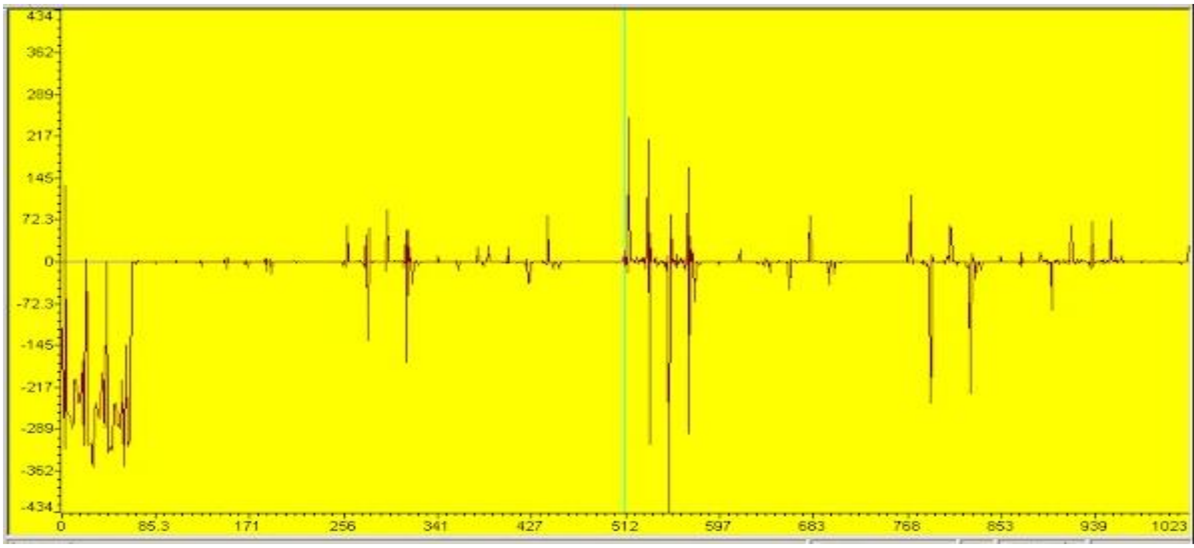

Figure 4.4(e): ECG mita record 100 with fourth level decomposition producing a_4 and d_4.

Figure 4.4 (a), (b), (c), (d) and (e) shows the original ECG signal along with the decomposed ECG signals of MIT-BIH record 100 and using Daubechies4 wavelet coefficients. The coefficients of details d_1, d_2, d_3, d_4 and approximation a_4 are retrieved and they are stored in the memory. Now further processing is done on these coefficients.

4.3 Coding of Decomposed Coefficients using BFP and RLE

The decomposed coefficients by DWT compressed using the RLE but before that they are passed through rounding process to reduce the truncation error and after that they are processed through hard thresholding process to discard the insignificant coefficients, which are below the threshold [65]. From these thresholded coefficients the Bits-of-Interest (BOI) are extracted and Significant Maps (SMs) are generated, then these SM's are divided into bytes and then these are compressed using Running Length Encoding (RLE) [69] and finally, the proper transmission packet size is designed for the transmission of compressed ECG signal. The overall compression scheme is as illustrated in block diagram in figure 4.5:

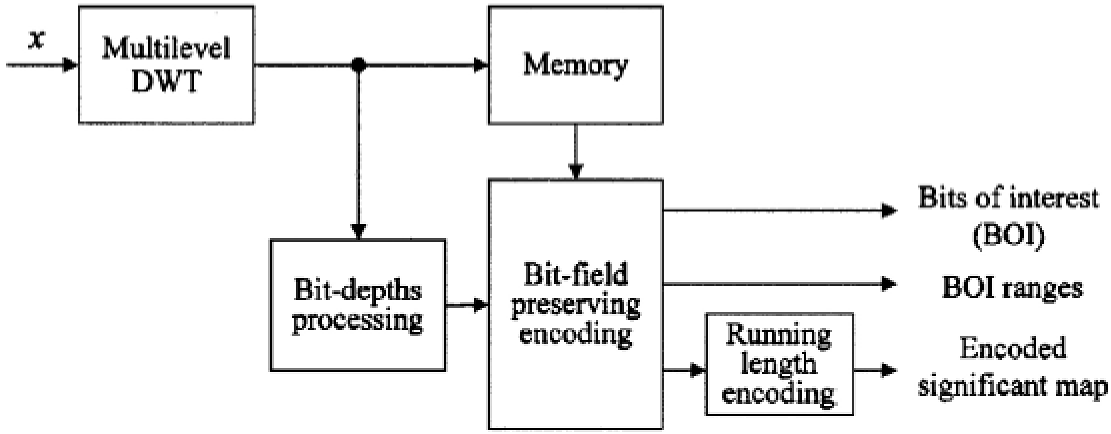

Figure 4.5: Block diagram for ECG compression based on Bit Field Preserving and Running Length Encoding.

Figure 4.5 [64] shows the block diagram for compression of ECG signal based on the Bit Field Preserving and Running Length Encoding.

4.3.1 The Bit Field Preserving

The bit field preserving coding is based on preserving the Bits of Interest (BOI) of the significant coefficients, i.e. coefficients above the threshold. To determine the BOI, there are two parameters, preserved bit length and the bit depth. The preserved bit length I_{Sb} is set by us according to the desired compression performance, where Sb stands for Sub-band $d_1, d_2, \ldots, d_J$ and so on up to a_J.

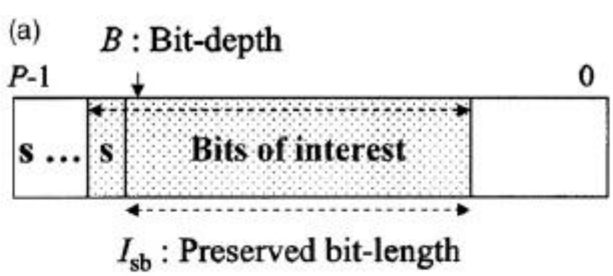

Figure 4.6(a): The bit manipulation of the bit-field preserving coding.

Figure 4.6(a) [64] shows how to calculate bit-depth and Bits of Interest based on the Preserved bit-length I_{Sb}. Bit-depth *B* is the most significant bit of the coefficient. BOI is calculated by taking I_{Sb} number of bits after the bit depth or most significant bit. A round-off mechanism is applied to improve the reduced precision. For example If six bits i.e. I_{Sb}=5 is used to extract the BOI of a coefficient say 254, only the underlined bits of $(00\underline{011111}110)_2$ are preserved, i.e. the modified value of the extracted BOI is 248 $(00\underline{011111}000)_2$ with a truncation error of 6. This truncation error can be reduced by applying a round-off mechanism, For this the original value is modified by adding 2^{B-ISb} , for this example which is equivalent to $(00000000100)_2$ before BOI extraction. After adding this round-off to the coefficient, The reconstructed value from the extracted BOI will now becomes 256 i.e. $(0\underline{100000}0000)_2$, so in this case the truncation error is reduced to 2. This example is illustrated in figure 4.6(b).

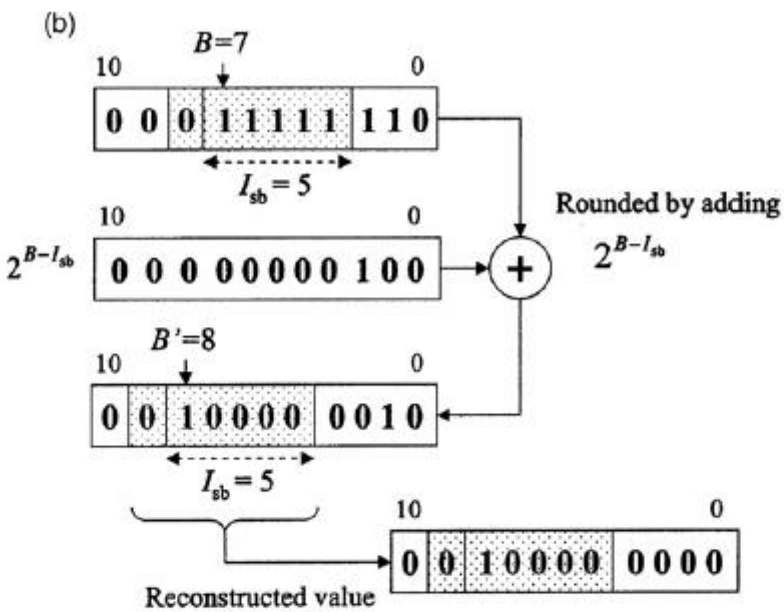

Figure 4.6(b): An example of the round-off mechanism

Figure 4.6(b) [64] briefly explain the operation of the round-off mechanism.

4.3.1.1 Bit Depth Processing:

In this step, the sub-band bit depth B_{Sb} is calculated for d_1, d_2,…., $d_{J,}$ a_J sub-bands. In this step, first of all the sub-band bit depth B_{Sb} is set to zero. When a new frame of ECG signal is ready for the data compression, with DWT operation, the bit depth B of the each coefficient is first determined. To reduce the truncation error to the coefficients, $2^{B\text{-}I_{Sb}}$ is added to each coefficient. Then the bit-depth (B') of the modified coefficient is determined, If the B' is greater than the current sub-band bit-depth $B_{Sb,}$ the B_{Sb} is modified to B', as all the frames of ECG signal is processed, this process will be continues and at the end, we will get the bit-depths B_{Sb} of each sub-band, so $B_{d1,}$ …., $B_{dJ,}$ B_{aJ} . To get the better performance result, the mean of the approximation signal a_4 is determined and each of the a_4 coefficient is subtracted by its mean before the BFP operation. This mean will be added back to the recovered a_4 coefficients during ECG decompression.

4.3.1.2 Thresholding:

Wavelet thresholding [70,71] is the signal estimation technique that exploits the capabilities of signal de-noising. There are two types of thresholding: Hard Thresholding and Soft Thresholding. Performance of thresholding will purely depend on the type of thresholding method and also the thresholding rule used for the given application. The hard threshold function tends to have bigger variance and it is unstable, because it is sensitive to even small changes in the signal. However, soft thresholding function is much more stable than hard thresholding and it tends to have a bigger bias due to the shrinkage of larger wavelet coefficients.

After determining the sub-band bit depths B_{Sb} of all the sub-bands, sub-band thresholds are first determined based on the desired preserved bit-length I_{Sb}. The sub-band threshold is given by,

$$Thres_{Sb}= 2^{B_{Sb}-I_{Sb}+1} \quad \text{……………..…………………………………}(4.3)$$

Now the wavelet coefficients that are stored in the memory are again retrieved and encoded. $2^{B_{Sb}-I_{Sb}}$ is again added to these coefficients to increase the accuracy of the compression. The coefficients are regarded as significant, if its modified value is greater than or equal to the computed threshold as given in equation (1). For each of the significant coefficients, a one is sent to the Significant Map (SM) and the BOI from $B_{Sb}+1$ to $B_{Sb}-I_{Sb}+1$ including the sign bit at $B_{Sb}+1$ is taken as output and for insignificant coefficients only a zero is sent to the SM.

The upper bit-field, i.e. $P-1$ to $B_{Sb}+2$ contains only the redundant sign bits so it is not considered. The lower bit-field, i.e. $B_{Sb}-I_{Sb}$ to 0 is also neglected in order to increase compression ratio (CR), but by doing this an undesired reconstruction error will occur, the round-off mechanism used can reduce this error.

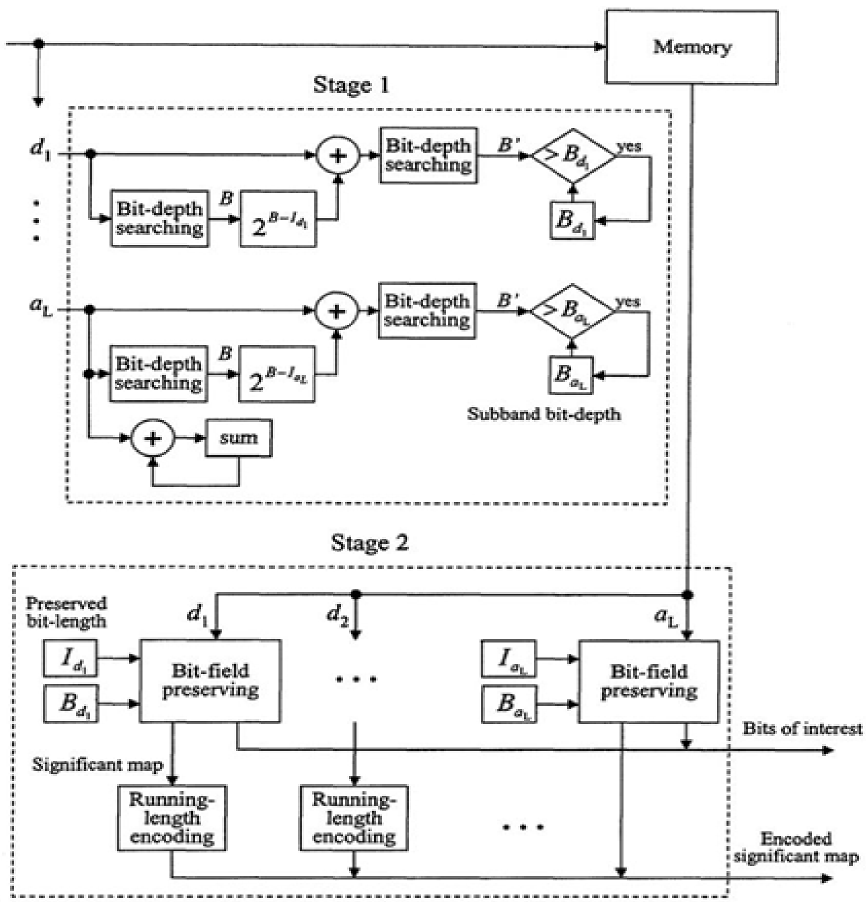

Figure 4.7: The two-stage processing of the bit-field preserving and Running Length Encoding (BFP-RLE).

Figure 4.7 [64] shows that the BFP is performed in two stages, the first stage is to compute the DWT and to determine the bit-depth of each sub-band and the second stage is to encode the wavelet coefficients. This BPF with thresholding is implemented on the C6713 DSP using C program. The flow chart of the Program is as follows:

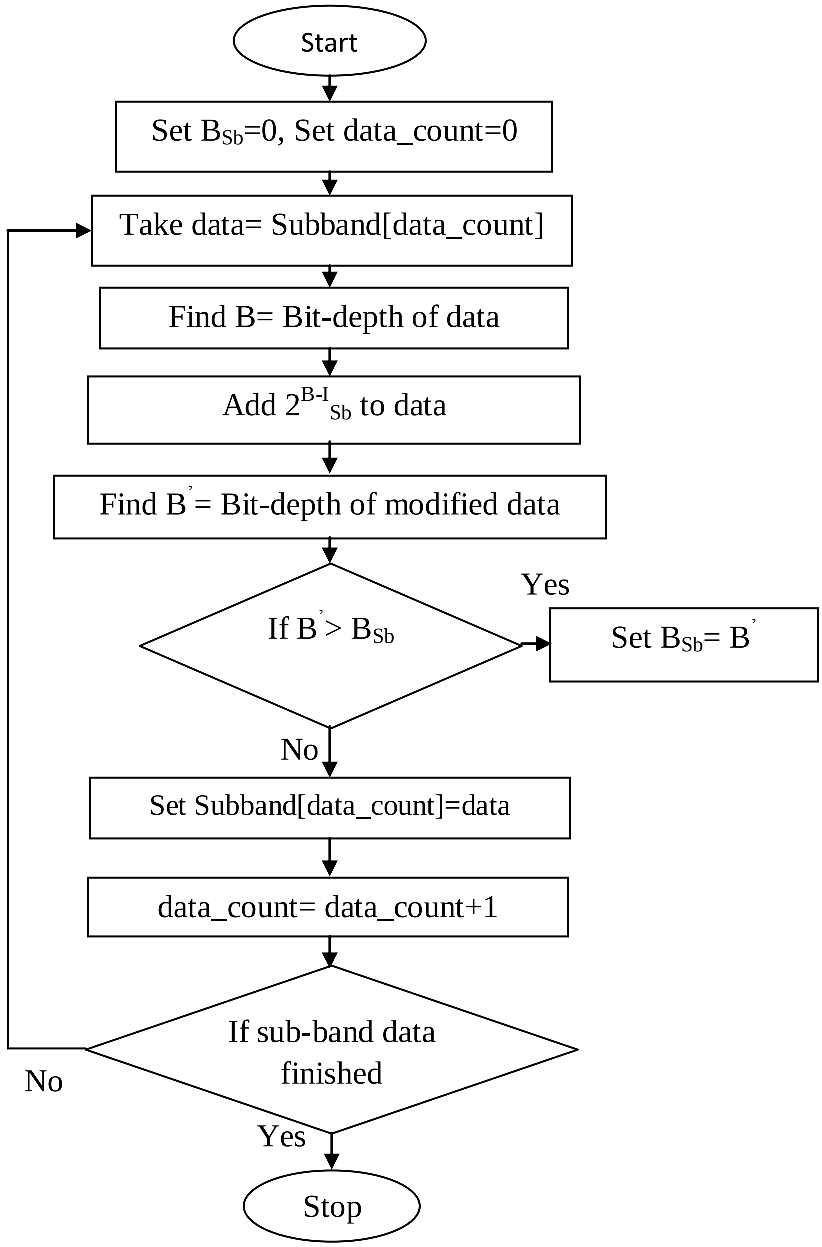

Figure 4.8: Programming flow diagram for Bit-Field-Preserving.

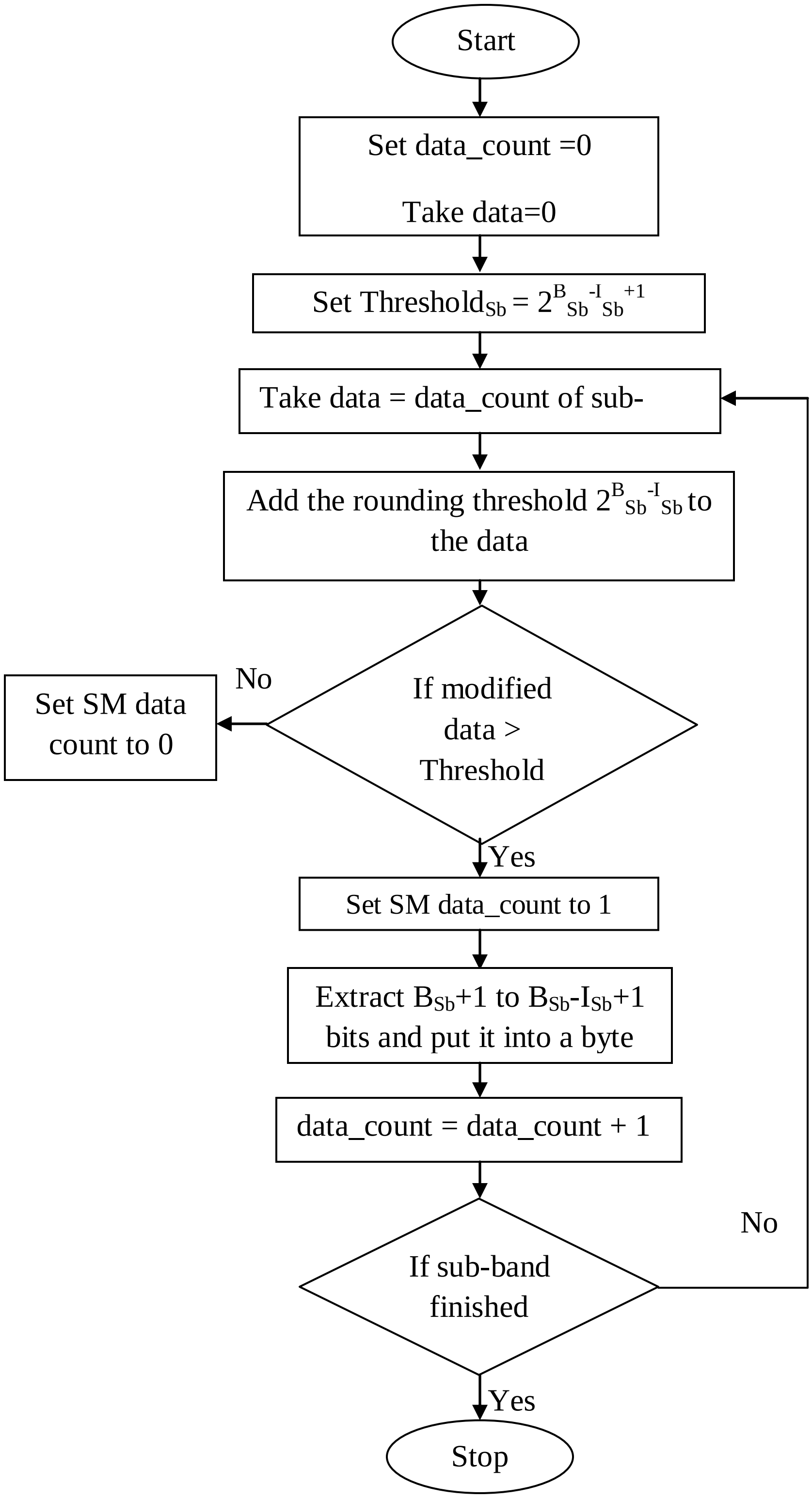

Figure 4.9: Programming flow diagram for Thresholding.

Figure 4.8 shows the programming flow chart for the BFP coding algorithm.

4.3.2 The Running Length Encoding

After BFP, a Significant Map (SM) bit stream is generated, which indicate the significance or insignificance of the wavelet coefficients, but with this the problem is that it will limit the compression performance. The Running Length Encoding [57,58,69,72] technique is applied to the SM bits.

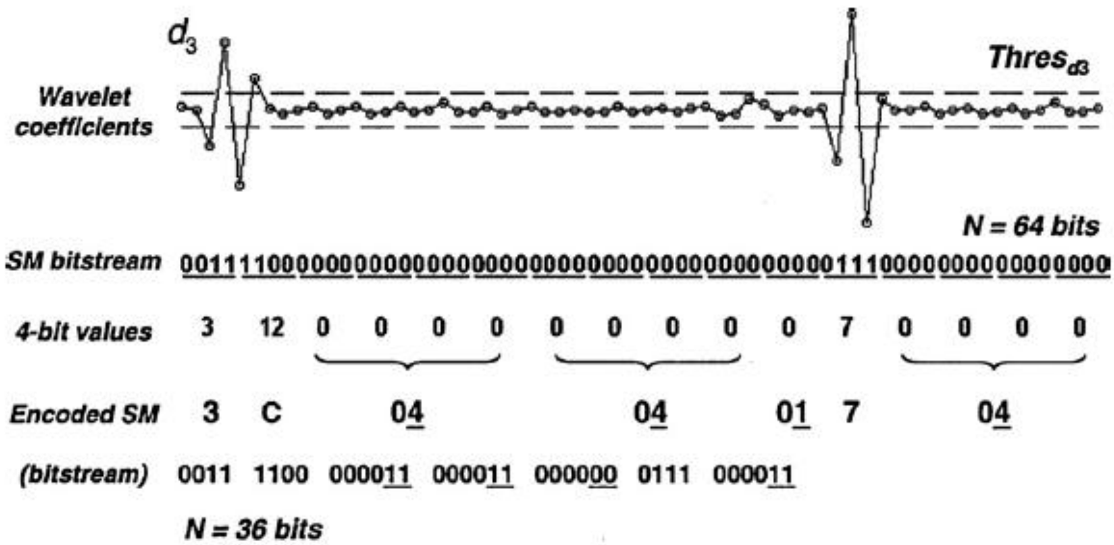

Figure 4.10: Example of the Running Length Encoding.

Figure 4.9 [64] illustrate the Running Length Encoding (RLE) manipulation on the SM bits of sub-band d_3. Several consecutive zeros are generated in the SM bit-stream between the adjacent QRS complexes. The SM bits are first transformed into the sequence of 4 bit values. If this 4 bit value is nonzero, then its data bits output are directly taken. If one or more of consecutive four bits are zero, then they are encoded as 0Nz. The first value 0 represents the zero and Nz represents the number of consecutive zeros. In the above example bit-length of Nz is set as 2, this means that upto 4 consecutive zeros can be encoded together. In above example after RLE, the number of SM bits are reduced from 64 to 36. The RLE method is not applied to the a_4

and d_4 sub-band coefficients, because they have fewer number of samples and also have fewer consecutive SM zeros.

This RLE method is also implemented on C6713 DSP using C programming language. The flow chart for the program is as follows:

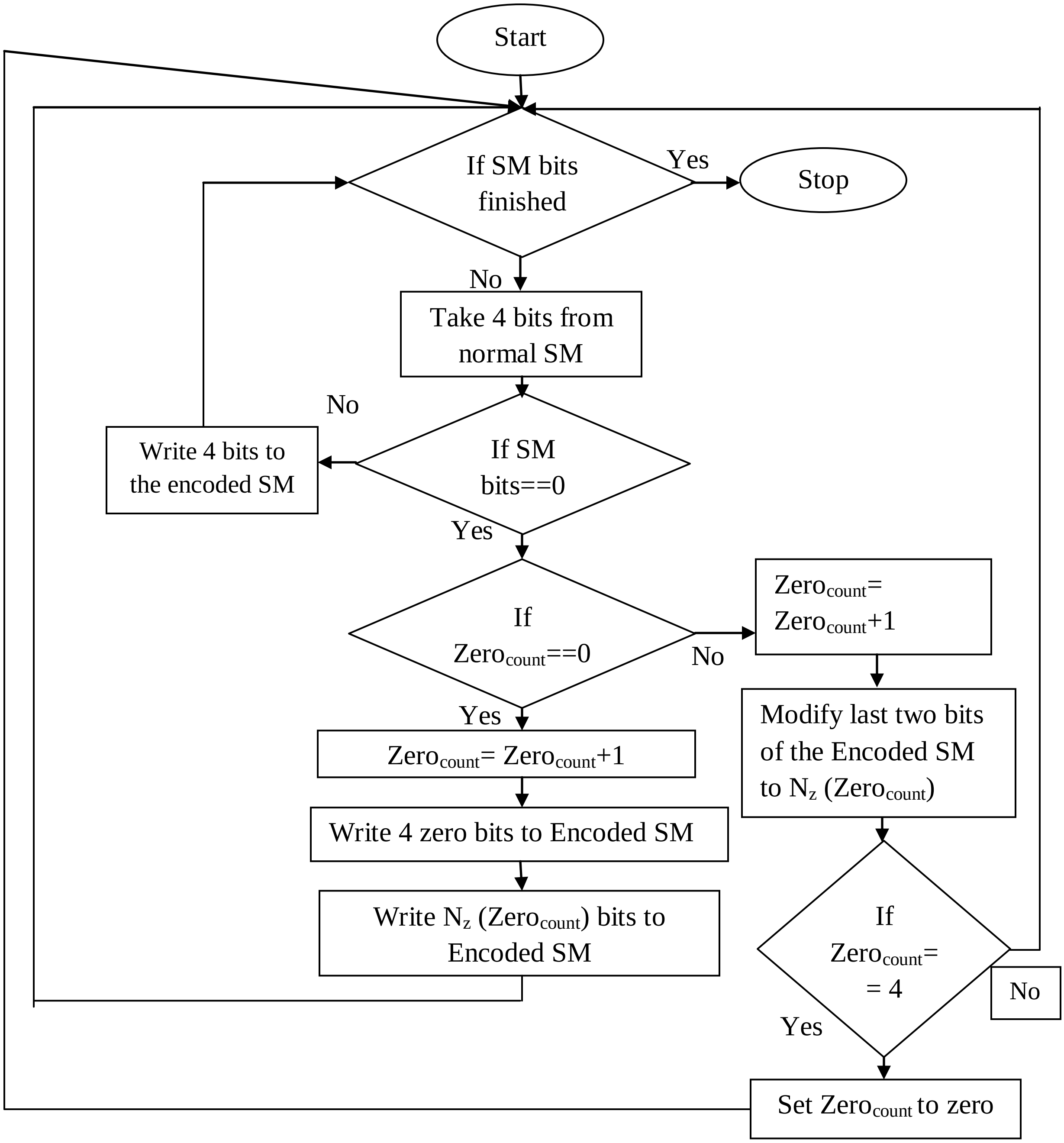

Figure 4.11: Running Length Encoding Program Flow Chart.

Figure 4.10 shows the flow diagram of Running Length Encoding Algorithm described above.

4.3.3 Packetizing the Compressed Data

To send the compressed BOI, BOI Range and SM packets, headers are required to indicate each segment of the compressed data. First, an indicator is used to indicate the total number of samples of the ECG signal Ns taken for compression at the beginning of the packet. Ns could have any value of 0, 1, 2, 3 and 4 which indicates that number of the compressed samples used is 64, 128, 256, 512 and 1024, respectively. Then, for each transformed sub-band by DWT, headers are created to indicate the Bits-of- Interest Range (BOI Range), the number of bytes used by the Bits-of-Interest (BOI Size), the number of bytes used by the significant map (Size SM) or the encoded significant map (Size SMe). The BOI and SM use the BOI Size and SM Size packets, respectively. Finally, the subtracted mean of the approximation sub-band (Mean of a_J) is divided into two bytes, which is placed at the end of the packet. At the receiver side, the packets are received in sequence, there they are decompressed after decoding them using the information arrived in headers.

All of the above compression algorithms are implemented on C6713 DSP using C source code. The proposed compression scheme is given in figure 4.11 [65]:

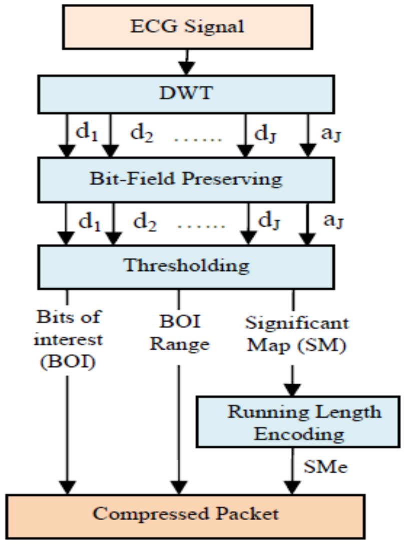

Figure 4.12: The proposed DWT-BFP-RLE compression scheme.

The experimental set-up is as shown in below figure:

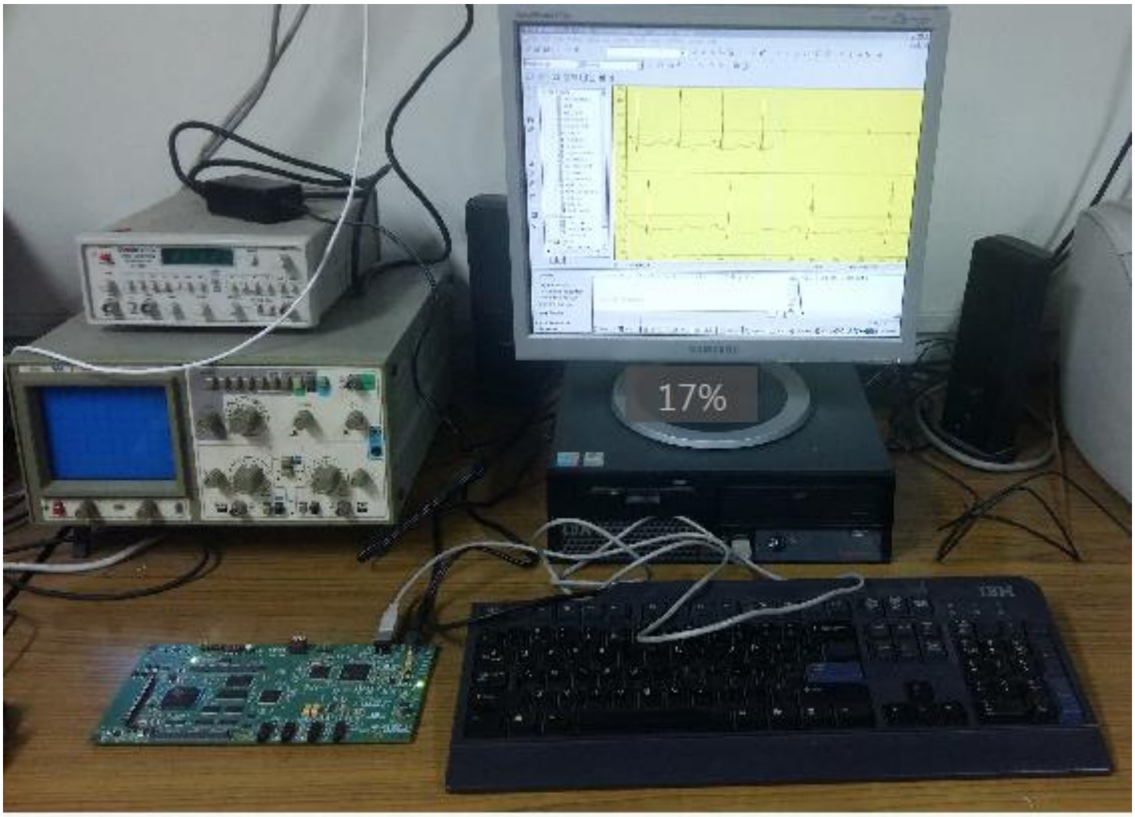

Figure 4.13: Experimental set-up for the proposed work.

4.4 ECG Decompression

At the receiver side, it is required to reconstruct the ECG signal from the received compressed ECG signal. For this exactly inverse operation of ECG signal compression is done. The reconstructed ECG signal may have some error with respect to the original ECG signal, If this error is zero, then it is called the perfect reconstruction of ECG signal, which is practically not possible. There will be some error associated with the reconstructed signal.

The block diagram of the Decompression of the ECG signal is shown in figure 4.13 [64]:

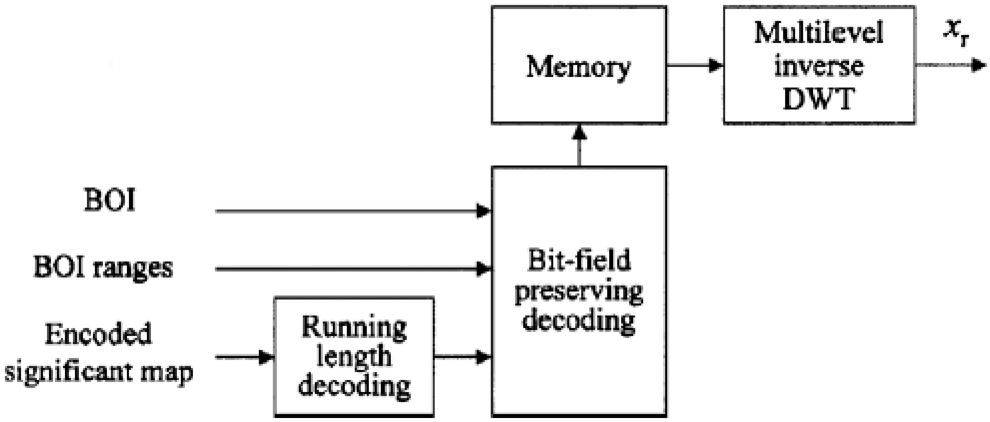

Figure 4.14: The Decompression block diagram

In decompression Encoded significant map obtained from Running Length Encoding is first decoded using Running Length decoding, and then DWT coefficients are reconstructed using the BOI and SM packets and finally the ECG signal is reconstructed from the DWT coefficients using the reconstruction filter banks.

This decompression is also implemented on the C6713 DSP using C programming. The flow chart for Running Length Decoding and Bit-Field Preserving Decoding are as follows:

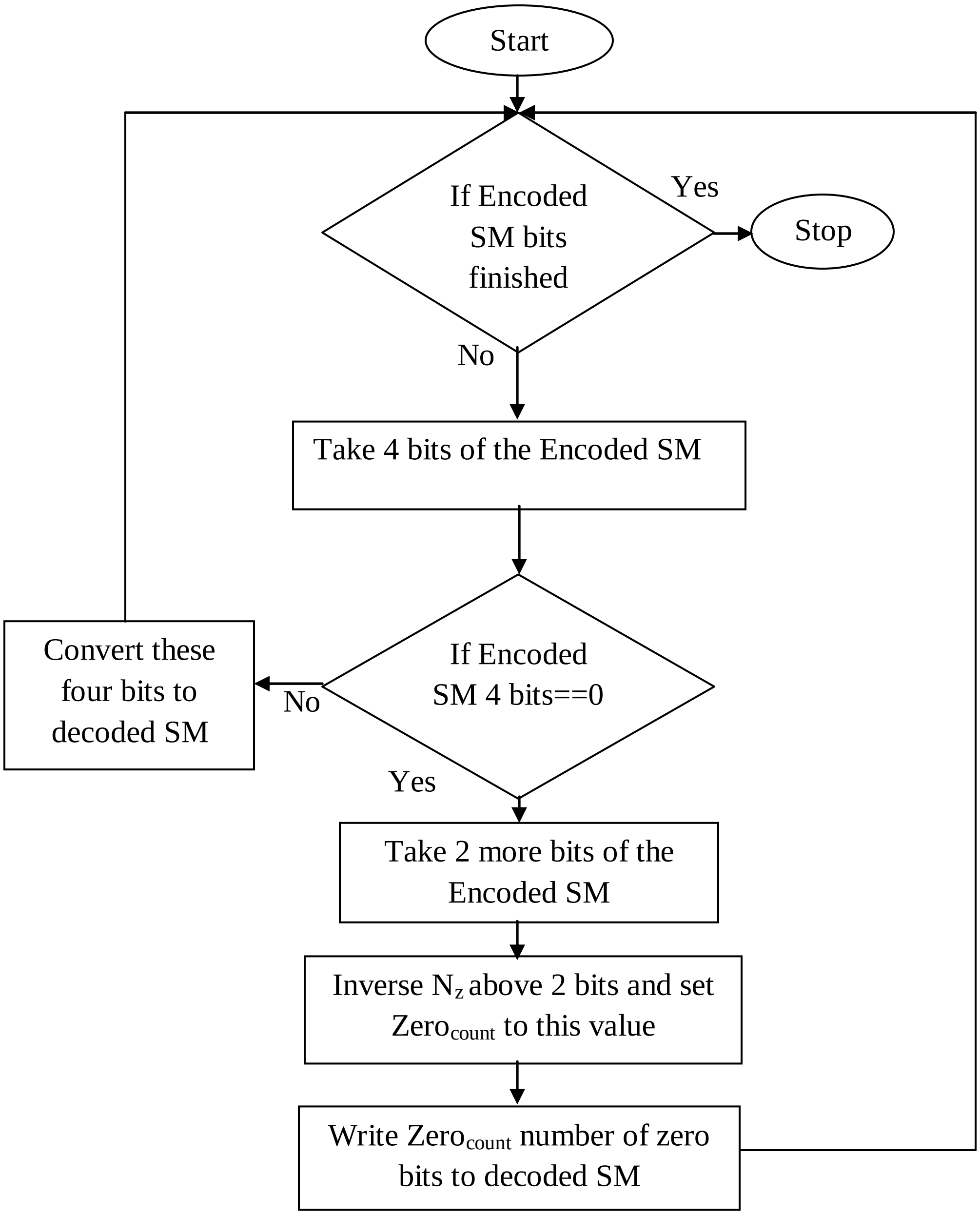

Figure 4.15: Programming flow chart for Running Length Decoding.

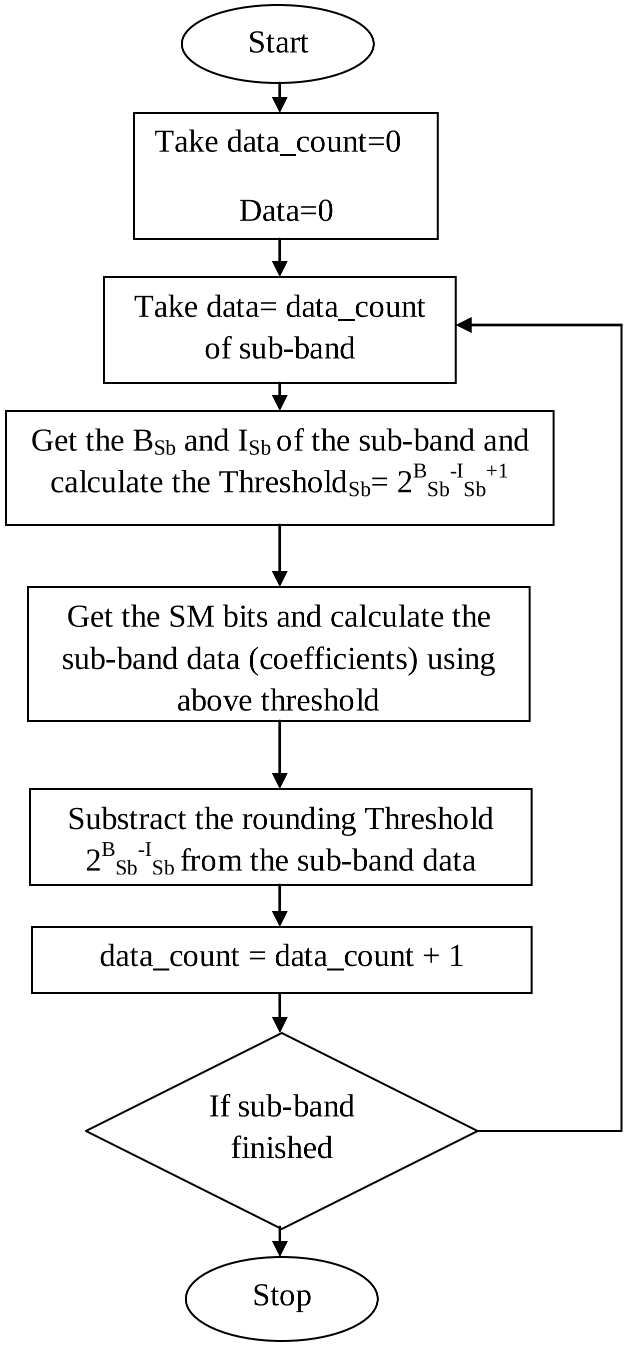

Figure 4.16: Programming Flow chart for inverse thresholding.

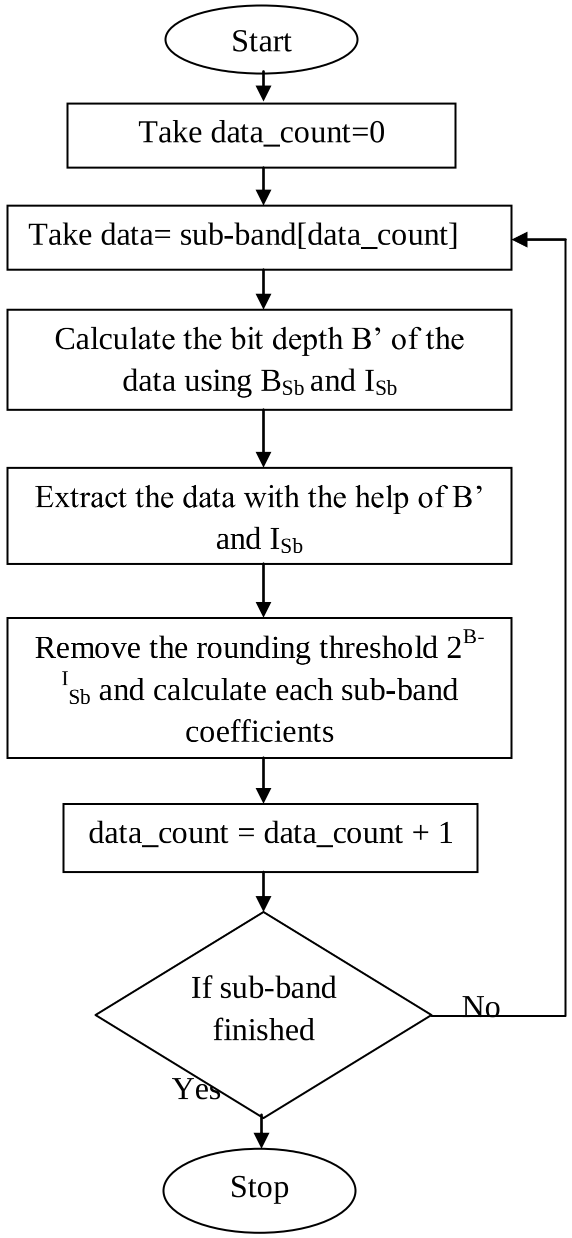

Figure 4.17: Programming flow chart for bit-field preserving decoding**.**

After this we will get the various sub-band (d_1,d_2,d_3,d_4,a_4) coefficients, the reconstruction of these is done with the help of inverse Mallat algorithm, which is just opposite of the forward Mallat algorithm. The reconstruction is as shown in the figure 4.17 [46], which is the reconstruction of the figure 4.3.

$$
\begin{bmatrix}
h_5 & & & & & h_0 & h_1 & h_2 & h_3 & h_4 \\
h_4 & h_5 & & & & & h_0 & h_1 & h_2 & h_3 \\
h_3 & h_4 & h_5 & & & & & h_0 & h_1 & h_2 \\
h_2 & h_3 & h_4 & h_5 & & & & & h_0 & h_1 \\
h_1 & h_2 & h_3 & h_4 & h_5 & & & & & h_0 \\
h_0 & h_1 & h_2 & h_3 & h_4 & h_5 & & & & \\
 & h_0 & h_1 & h_2 & h_3 & h_4 & h_5 & & & \\
 & & h_0 & h_1 & h_2 & h_3 & h_4 & h_5 & & \\
 & & & h_0 & h_1 & h_2 & h_3 & h_4 & h_5 & \\
 & & & & h_0 & h_1 & h_2 & h_3 & h_4 & h_5
\end{bmatrix}
\begin{bmatrix} a_1 \\ 0 \\ a_2 \\ 0 \\ a_3 \\ 0 \\ a_4 \\ 0 \\ a_5 \\ 0 \end{bmatrix}
+
\begin{bmatrix}
g_5 & & & & & g_0 & g_1 & g_2 & g_3 & g_4 \\
g_4 & g_5 & & & & & g_0 & g_1 & g_2 & g_3 \\
g_3 & g_4 & g_5 & & & & & g_0 & g_1 & g_2 \\
g_2 & g_3 & g_4 & g_5 & & & & & g_0 & g_1 \\
g_1 & g_2 & g_3 & g_4 & g_5 & & & & & g_0 \\
g_0 & g_1 & g_2 & g_3 & g_4 & g_5 & & & & \\
 & g_0 & g_1 & g_2 & g_3 & g_4 & g_5 & & & \\
 & & g_0 & g_1 & g_2 & g_3 & g_4 & g_5 & & \\
 & & & g_0 & g_1 & g_2 & g_3 & g_4 & g_5 & \\
 & & & & g_0 & g_1 & g_2 & g_3 & g_4 & g_5
\end{bmatrix}
\begin{bmatrix} d_1 \\ 0 \\ d_2 \\ 0 \\ d_3 \\ 0 \\ d_4 \\ 0 \\ d_5 \\ 0 \end{bmatrix}
=
\begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \\ s_8 \\ s_9 \\ s_{10} \end{bmatrix}
$$

Figure 4.18: Reconstruction from figure 4.3.

Figure 4.18 [46] shows the reconstruction written in an elegant format:

$$
\begin{bmatrix}
h_5 & g_5 & & & & & h_1 & g_1 & h_3 & g_3 \\
h_4 & g_4 & & & & & h_0 & g_0 & h_2 & g_2 \\
h_3 & g_3 & h_5 & g_5 & & & & & h_1 & g_1 \\
h_2 & g_2 & h_4 & g_4 & & & & & h_0 & g_0 \\
h_1 & g_1 & h_3 & g_3 & h_5 & g_5 & & & & \\
h_0 & g_0 & h_2 & g_2 & h_4 & g_4 & & & & \\
 & & h_1 & g_1 & h_3 & g_3 & h_5 & g_5 & & \\
 & & h_0 & g_0 & h_2 & g_2 & h_4 & g_4 & & \\
 & & & & h_1 & g_1 & h_3 & g_3 & h_5 & g_5 \\
 & & & & h_0 & g_0 & h_2 & g_2 & h_4 & g_4
\end{bmatrix}
\begin{bmatrix} a_1 \\ d_1 \\ a_2 \\ d_2 \\ a_3 \\ d_3 \\ a_4 \\ d_4 \\ a_5 \\ d_5 \end{bmatrix}
=
\begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \\ s_8 \\ s_9 \\ s_{10} \end{bmatrix}
$$

Figure 4.19: Altered reconstruction from figure 4.3.

The calculations done above can be easily implemented on the dsp board using circular buffer management, the reconstruction filters from the matrix in figure 4.17 will changed from:

$$\{h_0\ h_1\ h_2\ h_3\ h_4\ h_5\} \text{ and } \{g_0\ g_1\ g_2\ g_3\ g_4\ g_5\}$$

to

$$\{h_1\ g_1\ h_3\ g_3\ h_5\ g_5\} \text{ and } \{h_0\ g_0\ h_2\ g_2\ h_4\ g_4\}$$

This is done for convenience for implementing wavelets on C6713 DSP.

In the same way all the sub-bands are processed and reconstruction is done.

4.5 Compression Performance Evaluation:

In this work, a four-level (J = 4) lifting wavelet transform is adopted. The BFP encoding was respectively performed on the d_1, d_2, d_3, d_4 and a_4 coefficients. The numbers of data bits required for the decompression are as follows:

1) N_{BOI}, which represents the number of the BOI bits of significant coefficients.

2) N_{SM}, which represents the number of the encoded SM bits by the RLE;

3) N_{MBR}, which represents the number of bits to store the mean of $a4$ (11 bits) and the BOI ranges of J + 1 wavelet bands (=11 + (L+1)×2×4 bits).

The Compression Ratio (CR) is therefore given by,

$$CR = \frac{M \times I_{ECG}}{N_{BOI} + N_{SM} + N_{MBR}} \quad \text{...(4.4)}$$

where $M \times I_{ECG}$ represents the number of total bits of original ECG samples, $N_{BOI} + N_{SM} + N_{MBR}$ represents the number of the coded bits. The compression ratio and reconstruction quality of the ECG signal can be controlled by setting the preserved bit-length of each sub-band, because as the preserved bit-lengths increase, a better reconstruction quality can be achieved at the cost of CR due to the fact that in this case, more coefficients are regarded as significant and more bits are preserved. The reconstruction error is calculated in terms of the Percentage Root-mean-square Difference (PRD),which is defined as the normalized mean-square-error between original ECG $x(n)$ and the reconstructed ECG $x_r(n)$. PRD is given by,

$$PRD = \sqrt{\frac{\sum_{n=0}^{M-1}[x(n) - x_r(n)]^2}{\sum_{n=0}^{M-1}[x(n)]^2}} \times 100\% \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..(4.5)$$

There is one more performance parameter called Quality Score is there, which is the ratio between CR and PRD. It is given by,

$$QS = \frac{CR}{PRD} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(4.6)$$

The high Quality Score indicates a good compression performance.

4.6 Implementation Results and Discussions

4.6.1 Results

Here we have used, four different ECG records 100, 107, 117 and 119 to evaluate the compression performance. These records are widely used by the literatures to evaluate the compression performance. The proposed algorithm was simulated on C6713 Digital Signal Processor using C programming language.

The compression algorithm is evaluated by calculating the average CR, PRD and QS and their standard deviations using different preserved bit lengths I_{Sb}. Here format of I_{Sb} is $\{I_{d1}, I_{d2}, I_{d3}, I_{d4}, I_{a4}\}$, which represent the preserved bit-lengths in different sub-bands. We have taken the 1st 10 minutes record of each ECG signal and then this is divided into frames, each of which contains 1024 samples means 2.84 seconds of samples are taken into each frame. The CR is calculated as the ratio of the original ECG bits to the computed bits as shown by equation (4.1), original bits are fixed, i.e. 11bits×1024=11,264. for each frame results obtained for different preserved bit-lengths I_{Sb} are as follows:

mita Record	Compression Ratio (CR)	PRD(%)	Quality Score (QS)
	Preserved bit-length I_{Sb1}={1, 1, 2, 3, 6}		
100	8.821	1.287	6.854
107	8.375	1.755	4.772
117	9.069	0.988	9.179
119	9.054	6.072	1.491

Table 4.2: The Compression performance for I_{Sb}= {1,1,2,3,6}.

Mita Record	CR	PRD(%)	QS
	Preserved bit-length I_{Sb2}= {1, 3, 5, 5, 6}		
100	7.672	0.523	14.66
107	6.945	0.731	9.50
117	7.621	0.658	11.57
119	8.228	5.52	1.37

Table 4.3: The Compression performance for I_{Sb}={1,3,5,5,6}.

mita Record	CR	PRD(%)	QS
	Preserved bit-length I_{Sb3}= {1, 2, 5, 5, 6}		
100	8.074	0.541	14.93
107	7.115	0.871	8.16
117	7.712	0.647	11.91
119	8.281	5.782	1.432

Table 4.4: The Compression performance for I_{Sb}= {1,2,5,5,6}.

mita Record	CR	PRD(%)	QS
	Preserved bit-length I_{Sb4}= {1, 2, 3, 4, 6}		
100	8.341	0.858	9.721
107	7.838	1.063	7.371
117	8.800	0.751	11.710
119	8.752	5.990	1.460

Table 4.5: The Compression performance for I_{Sb}= {1,2,3,4,6}.

Mita record	CR	PRD(%)	QS
	Preserved bit-length I_{Sb5}={ 2, 4, 6, 7, 7}		
100	6.055	0.405	14.95
107	5.413	0.589	9.190
117	5.667	0.337	16.81
119	6.213	5.263	1.18

Table 4.6: The compression performance for I_{Sb}={2,4,6,7,7}.

Mita record	CR	PRD(%)	QS
	Preserved bit-length I_{Sb6}={ 1, 3, 5, 6, 6}		
100	7.419	0.507	14.633
107	6.378	0.715	8.920
117	7.234	0.645	11.215
119	7.865	5.496	1.431

Table 4.7: The compression performance for I_{Sb}={1,3,5,6,6}.

From table 4.2, 4.3, 4.4, 4.5, 4.6 and 4.7 the average value of CR, PRD and QS for each preserved bit-length I_{Sb} are as follows:

	CR_{avg}	PRD_{avg} (%)	QS_{avg}
I_{Sb1}={1,1,2,3,6}	8.829	2.520	5.574
I_{Sb2}={1,3,5,5,6}	7.616	1.858	9.275
I_{Sb3}={1,2,5,5,6}	7.790	1.960	9.108
I_{Sb4}={1,2,3,4,6}	8.433	2.165	7.560
I_{Sb5}={2,4,6,7,7}	5.837	1.648	10.532
I_{Sb6}={1,3,5,6,6}	7.224	1.841	9.049

Table 4.8: The average compression performance of all preserved bit-lengths.

From average results in Table 4.8 , we can conclude that I_{Sb1}={1,1,2,3,6} gives the best compression result but at the expense of poor reconstruction error (PRD). I_{Sb5}={2,4,6,7,7} gives the optimum value of CR and PRD and has highest Quality Score (QS).

The graphs for CR, PRD, QS and Average performance from above results are as follows:

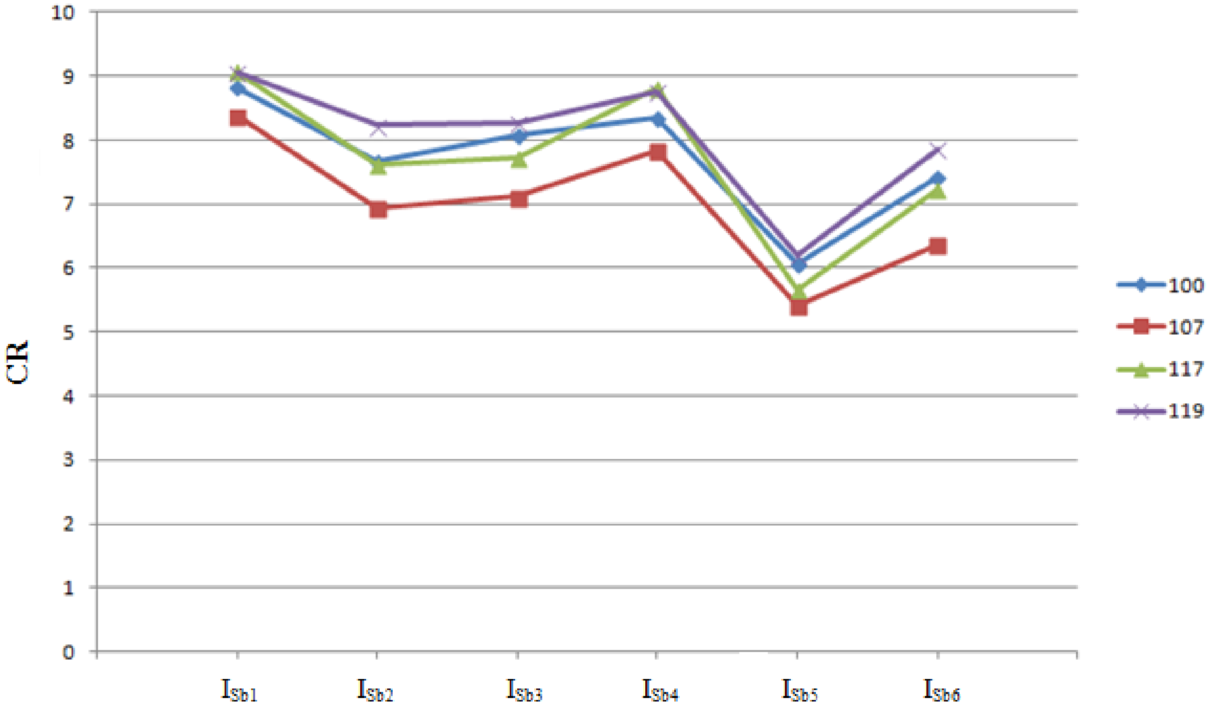

Figure 4.20: Average CR corresponding to different I_{Sb}.

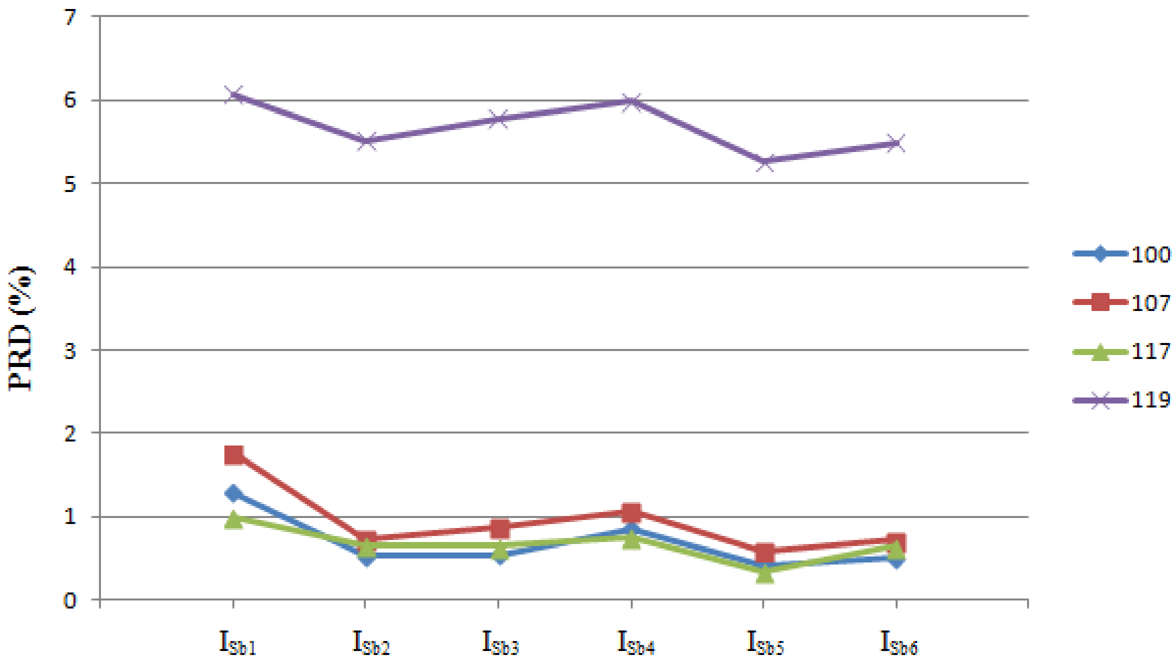

Figure 4.21: Average PRD corresponding to different I_{Sb}.

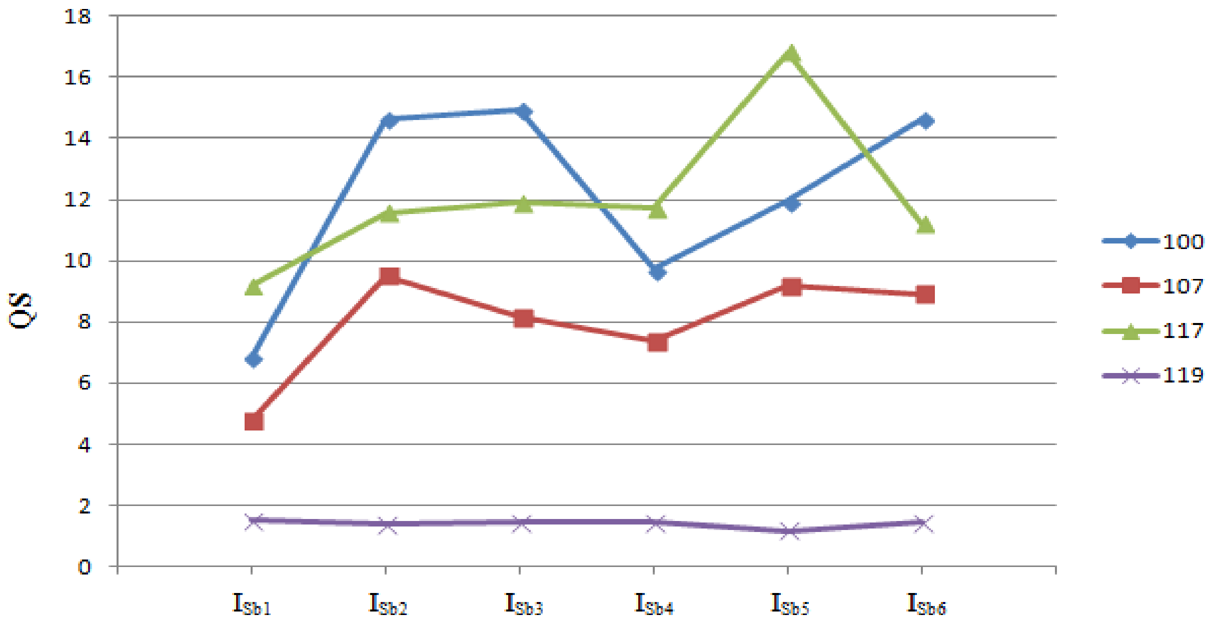

Figure 4.22: Average QS corresponding to different I_{Sb}.

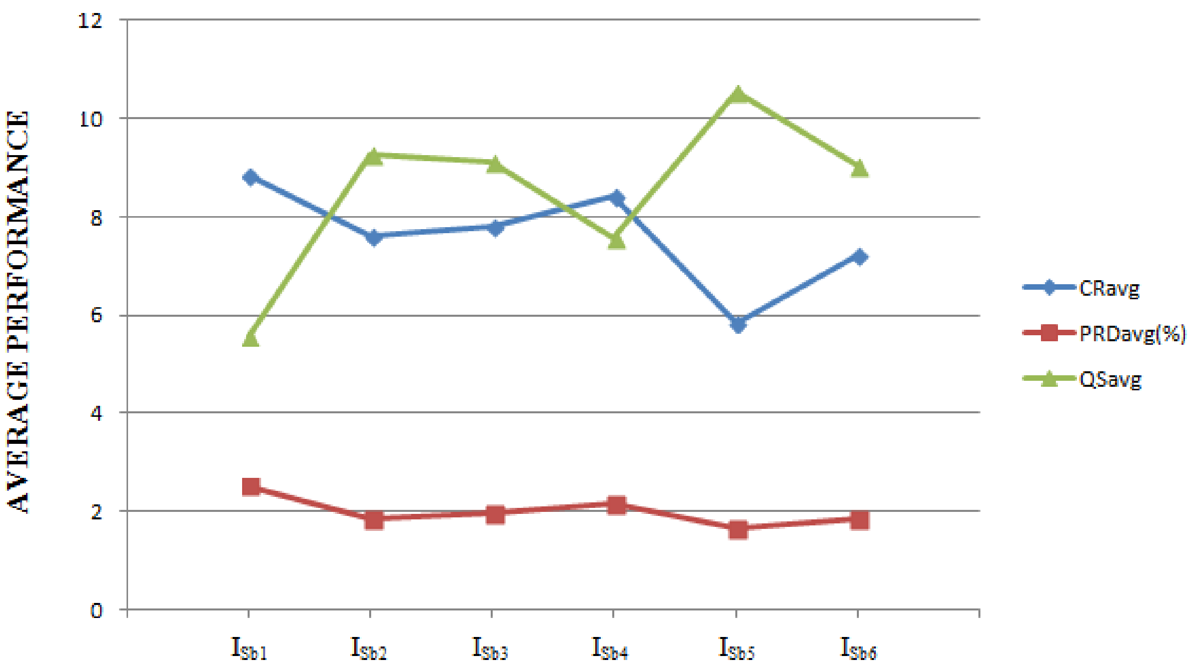

Figure 4.23: Average performance of different Preserved bit-lengths.

We can compare our results with other compression methods with same ECG signal mita record as follows:

Method	**mita record**	**CR**	**PRD(%)**	**QS**
SPIHT[11,28]	117	8:1	1.18	6.78
Hilton[73]		8:1	2.60	3.08
Dojhon[74]		8:1	3.90	2.05
With I_{Sb}={1,1,2,3,6}		9.069:1	0.988	9.179
With I_{Sb}={2,4,6,7,7}		5.667:1	0.337	16.81

Table 4.9: Performance comparison with other methods with N=1024 and duration 10 minutes for Db4 and J=4.

The graphs of original ECG signals, the reconstructed ECG signals along with error signals for different ECG records and for different I_{Sb}s for the first 1024 coefficients are drawn using matlab, which are as follows:

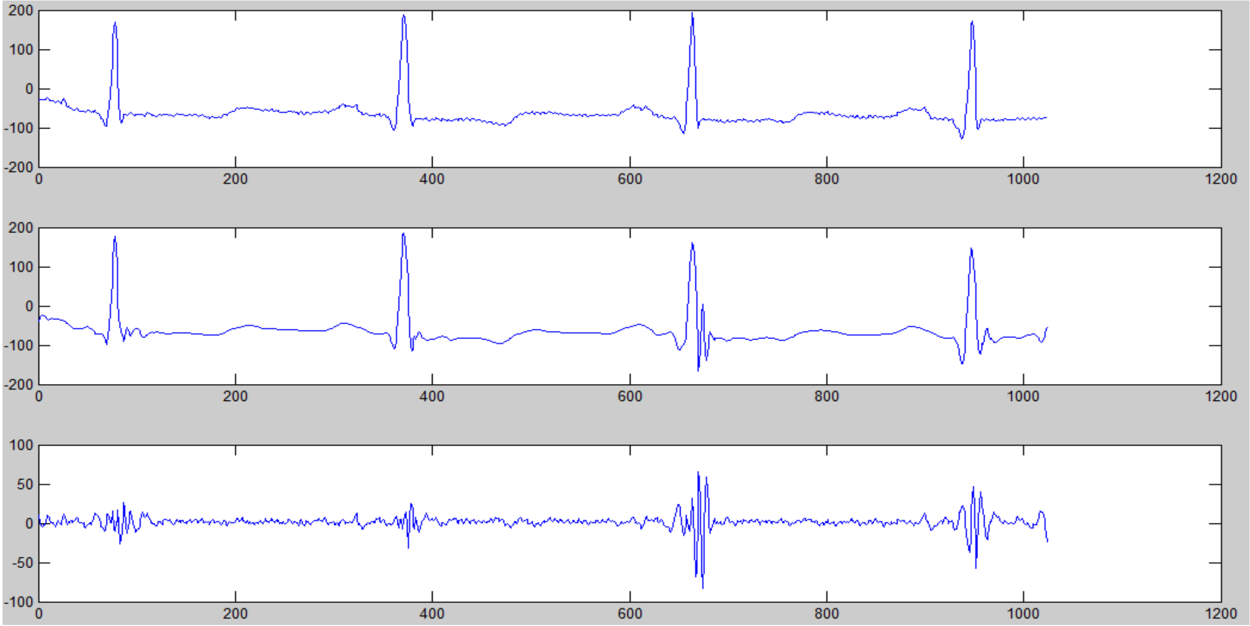

Figure 4.24: Compression results using I_{Sb}= {1, 1, 2, 3, 6}. From top to bottom, the original mita record 100, the reconstructed signal and the error signal.

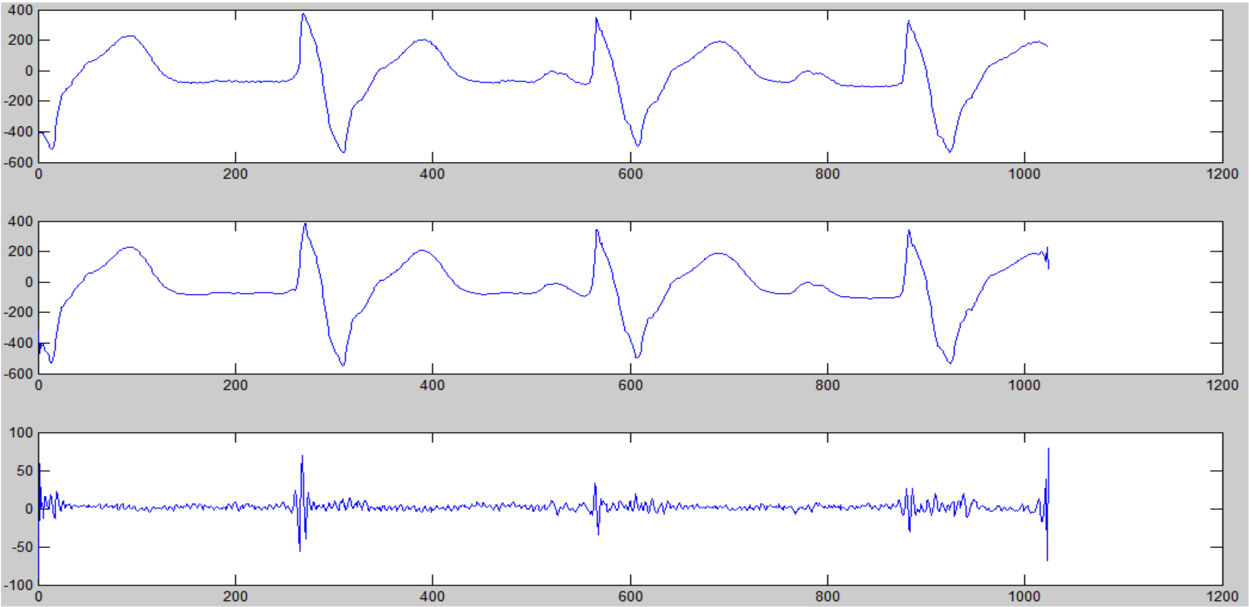

Figure 4.25: Compression results using I_{Sb}= {2,4,6,6,7}. From top to bottom, the original mita record 107, the reconstructed signal and the error signal.

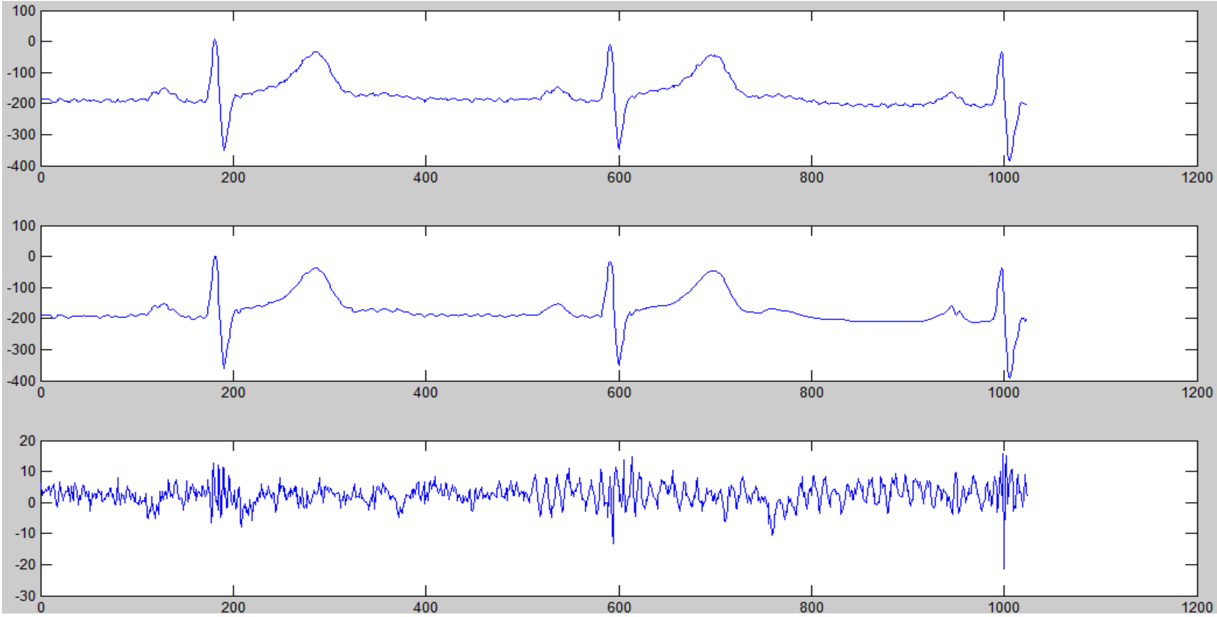

Figure 4.26: Compression results using I_{Sb}= {2,4,6,6,7}. From top to bottom, the original mita record 117, the reconstructed signal and the error signal.

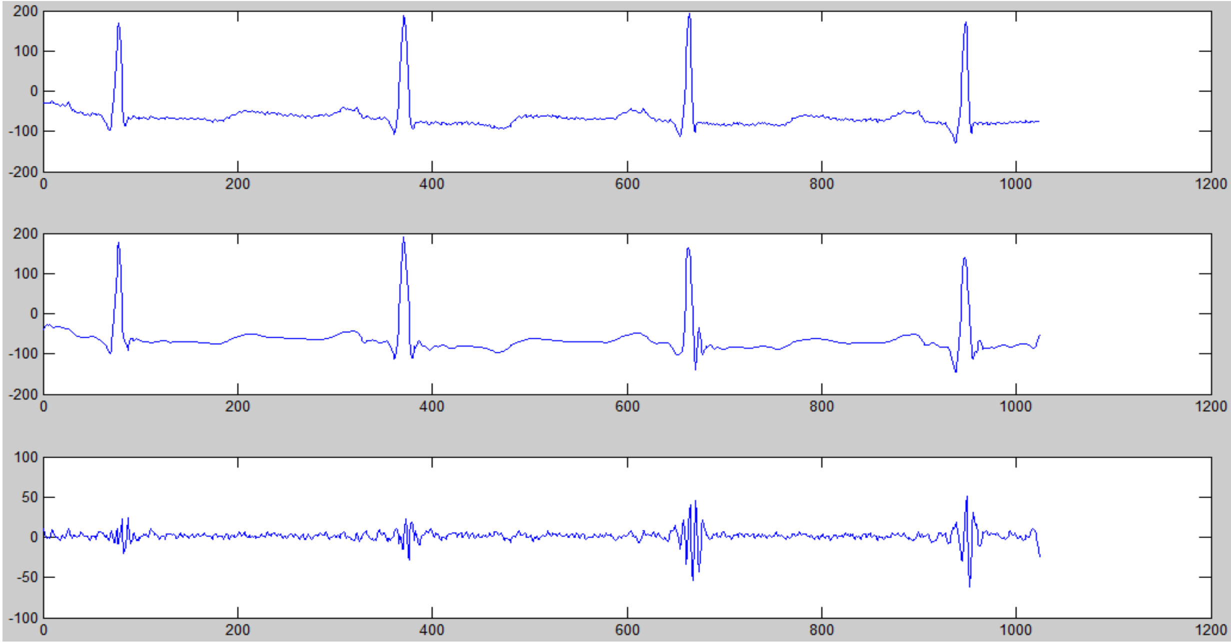

Figure 4.27: Compression results using I_{Sb}= {1,3,5,5,6}. From top to bottom, the original mita record 119, the reconstructed signal and the error signal.

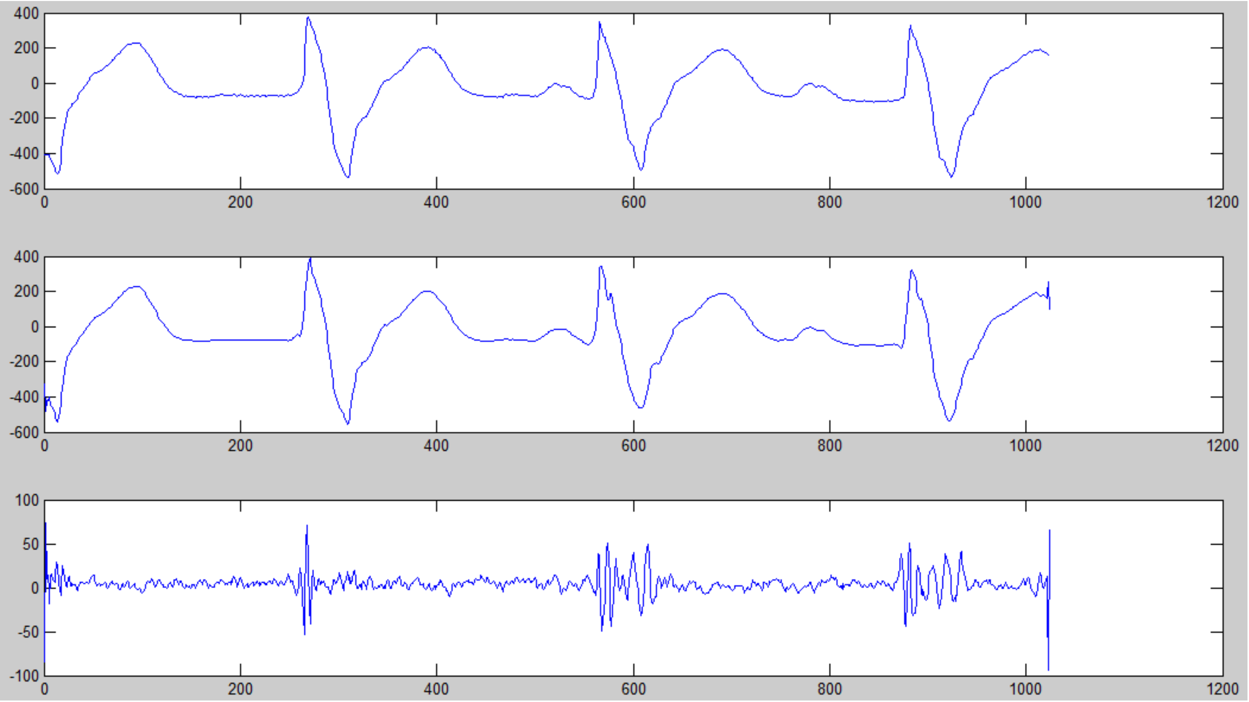

Figure 4.28: Compression results using I_{Sb}= {1,2,5,5,6}. From top to bottom, the original mita record 107, the reconstructed signal and the error signal.

4.6.2 Discussions

From tables 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8 and figures 4.20, 4.21, 4.23, 4.23, we can conclude that I_{Sb1}={1,1,2,3,6} gives the best compression performance among all the other I_{Sb}s and also with respect to the other compression method shown in table 4.6 but compromise is to be made with high reconstruction error (PRD) value, which leads to losing some diagnostic features such as QRS, P- or T-wave. By visual inspection of the graph (Figure 4.24 and 4.25), we can say that the the reconstruction error is high in ecg 100 record, while it is giving the good compression results. Thus we have to make a compromise between the quality of the signal, compression performance and number of compressed packets. After comparing the average of CR, PRD and QS for all of the six I_{Sb}s, we can conclude that I_{Sb}={2, 4, 6, 7, 7} gives the optimum value of the Quality Score and it provides the optimum solution for the Compression which maintaining permissible PRD. I_{Sb}={1,2,5,5,6} also give the good Quality score after I_{Sb}={2,4,6,6,7}.

When we compare figure 4.24 with I_{Sb}={1, 1, 2, 3, 6} and figure 4.25 with I_{Sb}={2, 4, 6, 6, 7}, by visual inspection we can see that, the reconstruction error in figure 4.24 is very high as compared to the reconstruction error in figure 4.25. So by visual inspection, we can say that PRD performance of I_{Sb}={2, 4, 6, 6, 7} is much better than the PRD performance of I_{Sb}= {1, 1, 2, 3, 6}. Also from figure 4.27 and 4.28, we can say that I_{Sb}= {1, 3, 5, 5 ,6} gives the better PRD performance than I_{Sb}= {1, 2, 5, 5, 6}. Hence these figures also verifying our results.

4.7 Summary

In this chapter I have presented the implementation of ECG compression using DWT-BFP-RLE technique on the TMS320C6713 DSP. First we have discussed that how to generate ECG signal using MIT-BIH arrhythmia database. After that we have discussed the decomposition of the ECG signal generated using Discrete Wavelet Transform. After that coding of the decomposed signal using BFP-RLE has been discussed. After that the reconstruction of the compressed ECG signal is discussed. We have discussed various performance parameters for compression analysis and at last comparison is done between different Preserved bit-length and also with some standard compression techniques. Graphs and reading of the various performance parameters have been presented.

Chapter 5

Conclusion and Future Scope of Work

5.1 Conclusion

In the present work ECG data compression based on DWT-BFP-RLE algorithm has been implemented using Texas Instrument's TMS320C6713 Digital Signal Processor. This work provides a real time implementation of ECG compression on a low power Digital Signal Processor to reduce the payload of the real-time ECG transmission. The proposed algorithm can control the block sizes of the compressed packets to fit into the available payload size. A compromise is made between the quality of the signal by increasing the Compression Ratio, Quality Score and decreasing the reconstruction error defined in terms of PRD and on the other side decreasing the number of the transmitted compressed packets. If the less number of packets are transmitted, the energy consumed by the transmitter antenna. This optimum number of packets with corresponding quality was selected by the preserved bit length factor used in the compression scheme. The visual inspection of the reconstructed signal and the average Quality Score value confirm that there is not much distorsion in the reconstructed ECG signal.

The study of the C6713 DSP along with its basic architecture and input/output operations has been done. The VLIW architecture along with Interrupts has also been studied. The Code Composer Studio used to program the DSP is also discussed.

The study of Discrete Wavelet Transform along with the Mallat algorithm has also been done. Study of ECG signal along with various other compression techniques has also been done. The BFP-RLE method for compression of the ECG signal has been studied. Various compression parameters for different ECG samples has been calculated.

We found that there are two ways to compresss the ECG signal using the BFP-RLE method, One is the to reduce the number of samples and their corresponding bit-fields for reconstruction of the data and other method is to utilize the RLE to compress the SM bits. The number of samples is reduced by considering only the significant coefficients. Instead of utilizing the whole bit field, only the BOI of significant coefficients are preserved. The negligence of the data bits beyond the BOI is same as the SPIHT algorithm, which also starts the encoding based on an initial threshold and stop when the desired compression performance is achieved. The negligence of the lower bit field results in the error in the reconstructed signal, the round-off mechanism can reduce this error.

5.2 Future Scope of work

The current implementation gives us good compression results using C6713 Digital Signal Processor but we have to make a trade-off between good compression and less reconstruction error. The better coding techniques can be implemented to obtain the good compression performance with less reconstruction error. Discrete Wavelet Packet Transform, which is a modification of traditional DWT, can also be used to increase the compression performance. ECG signal used in this thesis is sampled at 360 Hz and digitized in 11 bits, in future ECG records with different original sampling rate and different resolution levels can be tested. RTDX can also be used to add a link to MATLAB for control of entire algorithm.

Bibliography

[1] Tompkins W. J., “BIOMEDICAL DIGITAL SIGNAL PROCESSING”, PRENTICE HALL, 2000.

[2] Olkkonen J.,“DISCRETE WAVELET TRANSFORM - THEORY AND APPLICATIONS”, INTECWEB.ORG, 2011.

[3] Parak J., Havlik J., “ECG Signal Processing and Heart rate frequency detection methods” Department of Circuit Theory, Czech Technical University, Prague.

[4] Allen R. , “Biomedical Signal Processing and Control”, Elsevier, 2016.

[5] Van Fleet P.J., “Discrete Wavelet Transform- An Elementary Approach with Applications”, Wiley Interscience, 2007.

[6] G. Strang and T. Nguyen, “Wavelet and Filter Banks”, Wellesley- Cambridge Press, 1997.

[7] Guido R. C., “A note on a practical relationship between filters coefficients and scaling and wavelet function of the discrete wavelet transform”, Appl. Math. Lett., vol. 24, no. 7, pp.1257-1259, 2011.

[8] Abo-Zahhad M. , Rajoub B.A., “An effective coding technique for the compression of One-dimensional signals using wavelet transforms”, Elsevier Medical Engineering & Physics, 2002.

[9] Szilagyi S. M., Szilagyi L., David L., “ECG Signal Compression Using Adaptive Prediction”, 19th International Conference - IEEE/EMBS, Chicago, 1997.

[10] Cardenas-Barrera J.L., Lorenzo-Ginori J.V., “Mean-Shape Vector Quantizer for ECG Signal

Compression”, IEEE Transaction on Biomedical Engineering, Vol. 46, No. 1, January 1999.

[11] Lu Z., Kim Z.Y., Pearlman W.A., “Wavelet Compression of ECG signals by the Set Partitioning in Hierarchical Trees Algorithm”, IEEE Transactions on Biomedical Engineering, Vol. 47, No. 7, July 2000.

[12] Zigel Y., Cohen A. and Katz A., “ECG Signal Compression Using Analysis by Synthesis Coding”, IEEE Transactions On Biomedical Engineering, Vol. 47, No. 10, October 2000.

[13] Zigel Y., Cohen A. and Katz A., “The Weighted Diagnostic Distortion (WDD) Measure for ECG Signal Compression”, IEEE Transactions on Biomedical Engineering, Vol. 47, No. 11, November 2000.

[14] Arnavut Z., “ECG Signal Compression Based on Burrows-Wheeler Transformation and Inversion Ranks of Linear Prediction”, IEEE Transactions On Biomedical Engineering, Vol. 54, No. 3, March 2007.

[15] Sahrarian S.M.E., Fatemizadeh E., “Wavelet based 2-D ECG Data Compression Method Using SPIHT and VQ coding”, The International Conference on “Computer as a Tool”, EUROCON 2007.

[16] Fira C.M., Goras L., “An ECG Signals Compression Method and Its Validation Using Neural Networks (NNs)”, IEEE Transaction on Biomedical Engineering, Vol. 55, No. 4. April 2008.

[17] Wu T. C., Hung K.C., “Realization of RRO-NRDPWT-BASED ECG Signal Compression With Modified Run-Length Coding”, 2nd International Conference on Education Technology and Computer (ICETC), 2010.

[18] Ballesteros D. M., Moreno D. M., "FPGA Compression of ECG Signals by using Modified Convolution Scheme of the Discrete Wavelet Transform", Ingeniare, Revista chilena de ingenierfa, Vol. 20, No.1, 2012.

[19] Mishra A., Thakkar F., Modi C., "ECG Signal Compression using Compressive Sensing and Wavelet Transform", 34th Annual International Conference of the IEEE EMBS San Diego, California USA, 28 August -1 September, 2012.

[20] Ansari-Ram F., Hosseini-Khayat S., "ECG Signal Compression Using Compressed Sensing with Non-uniform Binary Matrices", The 16th CSI International Symposium on Artificial Intelligence and Signal Processing, AISP 2012.

[21] Tae-Hun Kim, Se-Yun Kim, Jeong-Hong Kim, "Curvature Based ECG Signal Compression for Effective Communication on WPAN", Journal of Communications and Networks, Vol. 14, No. 1, February 2012.

[22] Al-Busaidi A.M., Lazhar Khriji, "Digitally Filtered ECG Signal Using Low- Cost Microcontroller", IEEE Transactions, 2013.

[23] Surekha K.S, Patil B.P., "ECG Signal Compression using Hybrid 1D and 2D Wavelet Transform", Science and Information conference, London, UK, 2014.

[24] Abo-Zahhad, Abdel-Hamid, Mohamad M., "Compression of ECG signal based on DWT and exploiting the correlation between ECG Signal Samples", International Journal Communications, Network and System Sciences, 2014.

[25] Ukil A., Barlocher A., "Implementation of Discrete Wavelet Transform for Embedded Applications using TMS320VC5510", IEEE Transactions, 2014.

[26] Anamika Verma, Vasim Khan, "ECG Compression using Discrete Wavelet Transform with Coiflets and Daubechies Wavelets", International Journal of Electrical, Electronics and Computer Engineering, 2014.

[27] Ranjeet Kumar, Anil Kumar, "ECG Signal Compression Algorithm based on Joint Multiresolution Analysis (J-MRA)", IEEE Sponsored 2nd International Conference on Electronics and Communication System, ICECS 2015.

[28] Grafika Jati , Aprinaldi, "ECG Signal Compression by Predictive Coding and Set Partitioning in Hierarchical Trees (SPIHT)", ICACSIS, 2015.

[29] TMS320C6000 CPU and Instruction Set, SPRU 189F, Texas Instruments, Dallas, TX, 2000.

[30] TMS320C6000 Peripherals, SPRU190D, Texas Instruments, Dallas, TX, 2001.

[31] Modified Harvard Architecture,
http://en.wikipedia.org/wiki/modified_harvard_architecture

[32] Von Neumann Architecture,
http://en.wikipedia.org/wiki/Von_Neumann_architecture

[33] Texas Instruments, How to begin development today with the TMS320C6713 Floating-Point DSP, SPRA809.

[34] Texas Instruments, TMS320C6713B Floating Point Digital Signal Processor, SPRS294B.

[35] Rulph Chassaing, Digital Signal Processing and Applications with the C6713 and C6416 DSK, Wiley Inter-Science, 2005.

[36] Load/Store Architecture,
http://en.wikipedia.org/wiki/Load/Store_architecture.

[37] Tretter, Steven A., Communication System Design Using DSP Algorithms with Laboratory Experiments for the TMS320C6701 and TMS320C6711, Kluwer Academic, New York, 2003.

[38] Spectrum Digital, TMS320C6713 DSK Technical Reference, 2004.

[39] Texas Instruments, TMS320C6000 Programmer's Guide, SPRU198K.

[40] Texas Instruments, TMS320C6000 Assembly Language Tools User's Guide, SPRU186K, Dallas, TX, 2002.

[41] Texas Instruments, TMS320C6000 Optimizing C Compiler User's Guide, SPRU187K, Dallas, TX, 2002.

[42] Texas Instruments, Code Composer Studio User's Guide, SPRU328B.

[43] Texas Instruments, TLV320AIC23 Stereo Audio CODEC Data Manual, SLWS106C, July 2001.

[44] Texas Instruments, TMS320C6000 Chip Support Library API Reference Guide, SPRU401.

[45] Texas Instruments, TMS320C6000 DSP/BIOS, SPRU303B.

[46] Cesar lovescu, "Wavelet Transforms in the TMS320C55×", Application Report, SPRA800, January 2003.

[47] Mallat S. G., "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 2, No. 7, JULY, 1989.

[48] Zhao Hong-tu , Xi Dong-mei, “Analysis on Algorithm of Wavelet Transform And Its Realization in C Language”, Proceedings of the Third International Symposium on Electronic Commerce and Security Workshops (ISECS ’10), *China, pp. 336-338, July 2010.*

[49] Zhao Hong-tu, Yan Jing, “The Wavelet Decomposition And Reconstruction Based on The Matlab”, Proceedings of the Third International Symposium on Electronic Commerce and Security Workshops (ISECS ’10), China, pp. 143-145, July 2010.

[50] S. Mallat, “A wavelet tour of signal processing (second edition)”, Academic Press, 1999.

[51] Held G, Marshall TR., “Data and Image Compression: Tools and Techniques”, 4th ed. John Wiley and Sons, 1996.

[52] Indumathi K. “FIR Filter Design Using Discrete Wavelet Transform for ECG Compression”, International Journal of VLSI and Embedded Systems- IJVES, Vol. 04, Article 12201, Dec 2013

[53] Jalaleddine S.M.S., Hutchens C.G., Strattan R.D., Coberly W.A. “ECG data compression techniques — a unified approach” IEEE Transaction on Biomedical Eng. 1990.

[54] Djohan, A., Nguyen, T.Q., Tompkins, W.J., “ECG compression using discrete symmetric wavelet transform”, IEEE 17th Annual Conference, Engineering in Medicine and Biology Society, vol. 1, pp. 167-168, 1997.

[55] Abo-Zahhad, M., Ahmed, Sabah M., Al-Shrouf, A., Electrocardiogram data compression algorithm based on the linear prediction of the wavelet coefficients, Proc. of the 7th IEEE International Conference on Electronics, Circuits and Systems, vol. 1, Lebanon, Dec. 2000.

[56] Rajoub B., “An efficient coding algorithm for the compression of ECG signals using the wavelet transform”, IEEE Transaction Biomedical Engineering, 49 (2002) 233-239.

[57] Chouakri S. A., Benaiad M. M., Ahmed A.T., “Run Length encoding and wavelet transform based ECG compression algorithm for transmission via IEEE 802.11b WLAN channel”, 2009.

[58] Lee H. W., Hung K.C., “A Modified Run-Length Coding for the Realization of Wavelet-based ECG Data Compression System”, 2012.

[59] Patwari A. K., Singh V. P., “Analysis of ECG Signal Compression Technique Using Discrete Wavelet Transform for different wavelets”, International Journal of Engineering Trends and Technology (IJETT), Volume 8, No. 4, Feb 2014.

[60] Goudarzi M. M.. Taheri A., Pooyan M., “Efficient Method for ECG Compression Using Two Dimentional Multiwavelet Transform”, International Journal of Information Technology, Vol. 2, No. 4, 2009.

[61] Ciocoiu I.B., “ECG Signal Compression Using 2D Wavelet Transform”, International Journal of Advanced Science and Technology”, Vol. 13, Dec 2009.

[62] The MIT/BIH Arrhythmia Database, Massachusetts Institute of Technology, MA,1997.

[63] MATLAB, www.mathworks.com/dsp-system-toolbox/examples.

[64] Chan H.L., Siao Y. C., Yu S. F., “Wavelet based ECG compression by bit-field preserving and running length encoding”, Elsevier computer and programs in biomedicine 90 (2008) 1-8.

[65] Al-Busaidi A.M., Touati F., “Real-Time DWT-Based Compression for Wearable Electrocardiogram Monitoring System”, Proceedings of the 8th IEEE GCC Conference and Exhibition, Muscat, Oman, February, 2015.

[66] P.S. Addison, “The Illustrated Wavelet Transform Handbook: Introductory theory and Applications in Science, Engineering, Medicine and Finance”, CRC, 2002.

[67] Manikandan M. S., Dandapat S., “Wavelet based Electrocardiogram signal compression methods and their performance a prospective review”, Biomedical Signal Processing and Control, 2014, vol. 14, pp. 73-107.

[68] MATLAB Wavelet Toolbox, www.mathworks.com/products/**wavelet**.

*[69] Lee H.W., Hung K.C., “*A Modified Run-Length Coding towards the Realization of a RRO-NRDPWT-Based ECG Data Compression System”, EURASIP Journal on Advances in Signal Processing, Vol. 2011, Article 703752, March 2011.

[70] Karthikeyan P., Murugappan M., and Yaacob S., “ECG Signal Denoising Using Wavelet Thresholding Techniques in Human Stress Assessment”, International Journal on Electrical Engineering and Informatics - Volume 4, Number 2, July 2012.

[71] Georgieva-Tsaneva G., Tcheshmedjiev K., “Denoising of Electrocardiogram Data with Methods of Wavelet Transform”, International Conference on Computer Systems and Technologies,2013.

[72] Joseph S., Srikanth N., Abhilash J.E., “A Novel Approach of Modified Run Length Encoding Scheme for High Speed Data Communication Application”, International Journal of Science and Research (IJSR) ,ISSN (Online): 2319-7064.

[73] Hilton M.L., “Wavelet and Wavelet Packet compression of electrocardiograms,” IEEE Transaction on Biomedical Engineering, vol. 44, no. 5, pp. 394-402, 1997.

[74] Djohan A., Nguyen T.Q. and Tompkins T.J., “ECG compression using discrete symmetric wavelet transform,” IEEE 17th Annual Conference, Engineering in Medicine and Biology Society, vol. 1, pp. 167-168, 1997.

This is Rahul Chandwani, a Final year Master degree student(M.E. , Communication Engineering) of Jadavpur University. I have passed my Bachelor Degree (B.E. in Electronics & Telecommunication Engineering) from Bhilai Institute of Technology, Durg, Chhattisgarh. I have passed my High School and Higher Secondary from Bharat Mata Higher Secondary School, Bilaspur, Chhattisgarh . My present research focuses on ECG signal compression using DWT-BFP-RLE algorithm on TMS320C6713 Digital Signal Processor.

Mobile :9038774011
Mail ID : rah228758@gmail.com.