Implementation of Low Complexity Advanced Audio Codec

Ms. Sabale Amruta V and Dr. Sarwade Nisha

ABSTRACT: The audio compression has become the need of day to day life. The compression algorithms help to reduce the bandwidth requirements and also provide a level of security for the data being transmitted. This paper reviews implementation of the encoder and decoder of Low Complexity Advanced Audio Codec (LC-AAC). In the encoder section audio is compressed using Psychoacoustic Model (PAM). Further compression is done with source coding techniques like, Run Length Coding and Huffman Coding. Decoder decompresses the audio with Huffman decoding and Run Length Decoding and tries to get the approximately original audio.

KEYWORDS— AUDIO COMPRESSION, LOSSY AND LOSSLESS DATA COMPRESSION, MP3, ADVANCED AUDIO CODING (AAC), AAC+.

I: INTRODUCTION

Audio technology is an important part in our life. Every day, people listen to their favorite songs from broadcast, iPod, cellular phone etc. The applications of digital audio technique include broadcast system, portable players, iPod, and mobile phone…etc. Since MP3 has been published, and became popular consumer applications, the digital audio technique is an important part in daily life.

After the MP3 became very popular in the world, the organization of Moving Picture Experts Group (MPEG) proposed MPEG-2 Advance Audio Codec (AAC) to be the next generation audio standard. Both the performance and compression ratio of AAC are better than MP3. However, the algorithm is more complex and computation-intensive. Psychoacoustic Model (PAM) is the key component in the MPEG-2 AAC encoder. AAC gives better signal to noise ratio and has lower bit-rate as compare with MP3.

This thesis report reviews implementation of the encoder and decoder of Low Complexity Advanced Audio Codec (LC-AAC). It contains encoder and decoder sections. In the encoder section audio is compressed using Psychoacoustic Model (PAM). Further compression is done with source coding techniques like, Run Length Coding and Huffman Coding. Decoder decompresses the audio with Huffman decoding and Run Length Decoding and tries to get the approximately original audio.

All the blocks of Low Complexity Advanced Audio Coder are implemented using MATLAB simulator version 9 (MATLAB R2009a). High quality audio compression has found its way in many applications. Early research on audio has shifted to standardization efforts of ISO/IEC and ITU-R 10 years ago. In the last couple of years, Internet audio broadcasting has come in powerful category of this type of high quality applications. These techniques become more and more popular in many parts of the world because of the business for the music industry.

Ms. Sabale Amruta V is a M.Tech.[Electronics and Telecommunication] Student, Department Of Electrical Engineering, VJTI, Mumbai, Maharashtra and Dr. Sarwade Nisha is working as Asst. Professor, Department Of Electrical Engineering, VJTI, Mumbai, Maharashtra, Email: amrutas1990@gmail.com, npsarwade@vjti.org.in

This paper is organized as follows. In Section II, An Advanced audio coder overview is discussed. In Section III, AAC encoder and decoder details are given. In section IV, PAM details are described. Section V gives simulation results. Finally, conclude this paper in Section VI.

II: ADVANCED AUDIO CODEC-AN OVERVIEW

The tools use in MPEG-2 AAC encoder includes:

- Gain control
- Psychoacoustic model
- Filter Bank
- Prediction
- Quantization and Coding
- Noiseless Coding
- Bitstream Multiplexing
- Temporal Noise Shaping (TNS)
- Mid/Side (M/S) Stereo Coding
- Intensity Stereo Coding

In order to provide different kind of application, the AAC standard offers three profiles. Those profiles are main profile, Low-Complexity (LC) profile, and Scalable Sampling Rate (SSR) profile.

Tools	Main	**LC**	SSR
Noiseless coding	Yes	**Yes**	Yes
Quantization/Coding	Yes	**Yes**	Yes
M/S	Yes	**Yes**	Yes
Prediction	Yes	**No**	No
Intensity/coupling	Yes	**No**	No
TNS	Yes	**Limited**	Limited
Filter bank	Yes	**Yes**	Yes
Gain control	No	**No**	Yes

Table 1: Tools usage for different profile

III: MPEG-2 LC-AAC Encoder

The block diagram of the LC-AAC encoder is shown in Figure 1. The tools use in LC-AAC encoder includes:

- Psychoacoustic model (PAM)
- Filter Bank
- Temporal Noise Shaping (TNS)
- Quantization and Coding
- Bit-stream Multiplexing

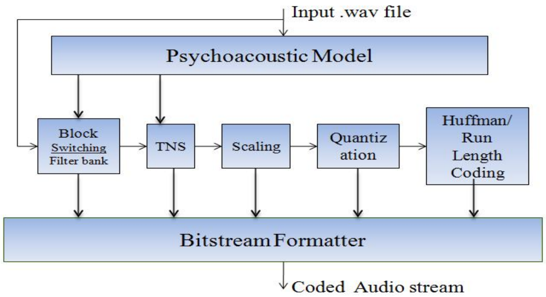

Figure 1: **Block Diagram of LC-AAC**

1.Psychoacoustic Model (PAM)
The psychoacoustic model (PAM) is the most important component of perceptual audio coding. It is used to model human's sense of hearing system to filter the signal that human can't recognize. There are two psychoacoustic models used in MPEG-Audio standards. Psychoacoustic model 1 has been used in MPEG-1 Layer 1 and Layer 2, while model 2 is used in Layer 3 and later. *The psychoacoustic model 2 is better for simulating human system but it takes the cost on heavy computation compared to model 1.* The psychoacoustic model analysis has two tasks to execute in MPEG audio encoder, decide what block type to use for filterbank outside of PAM and calculating the signal to mask ratio.

2.Windowing, Block Switching and Filtering
Window Shape Adaptation
In order to get better spectral separation ability, AAC supports two different window shapes that can be switched dynamically. They are Kaiser-Bessel derived (KBD) window and sine shaped window. The KBD window achieves better stop band attenuation while the sine window has better pass band selectivity.

3.Block Type Decision
In order to reduce pre-echo effect shown in Figure 2, the AAC supports different block sizes for MDCT. In the encoder, the filter-bank takes the appropriate block of time samples, modulates them by an appropriate window function, and performs the MDCT. Each block of input samples is overlapped by 50% with the immediately preceding block and the following block. The transform input block length N can be set to either 2048 (Long block with 50% overlapping) or 256 (Short block with 50% overlapping) samples. Because of the transform, filter-bank will get more resolution in frequency domain when the block size is longer. Oppositely, the transform of filter-bank will get more resolution in time domain when the block size is shorter. When the audio signal is stationary, the long block is used for encoding. When the signal is transient, the short block is used for encoding to avoid the pre-echo effect. While the short block is used, there is the some overhead on calculation and side information compared to long block size.

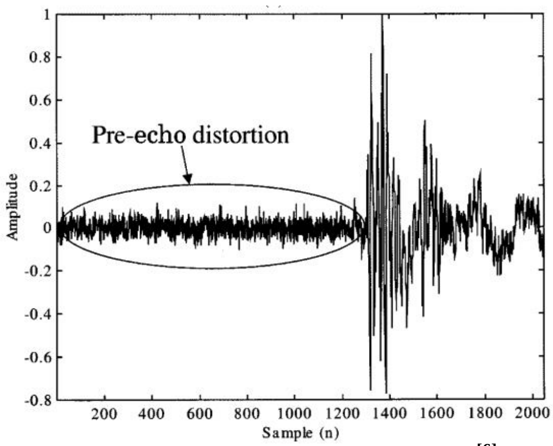

Figure 2 Pre-echo Example [6]

In AAC standard, it has four block (LONG WINDOW, SHORT WINDOW, START WINDOW, and STOP WINDOW) support shown as Figure 3 and in Figure 4 and uses the PE which is estimated by PAM to determine the signal property. The result of the PE is used to decide block switch mechanism. The START WINDOW and STOP WINDOW are buffers when the block type is changed from LONG to SHORT or SHORT to LONG, respectively.

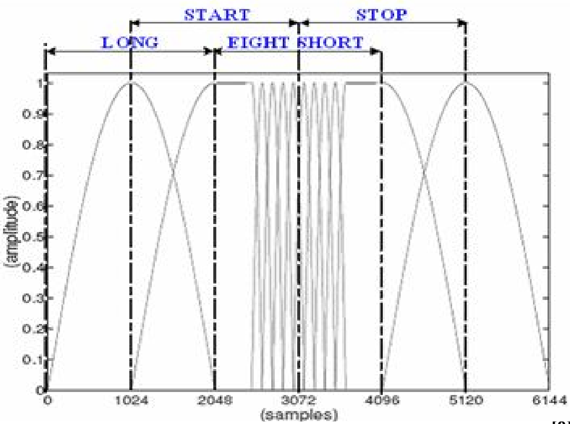

Figure 3 Four windows used in LC-AAC [8]

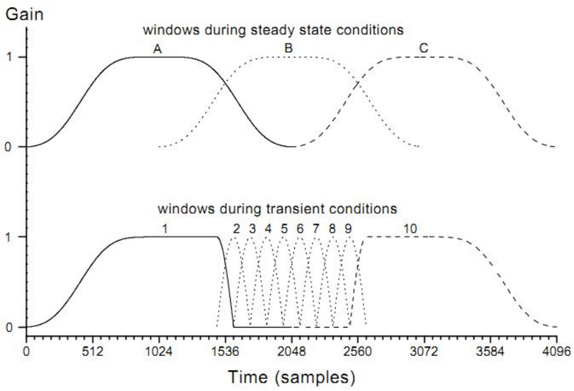

Figure 4 Example of block switching during transient signal condition [4]

4.Modified Discrete Cosine Transform (MDCT)
The MPEG-2 AAC standards and the later versions of new standard use the MDCT of longer length to increase the resolution of spectrum. For the time frequency transforms, most of the present day audio coders use modified cosine transform (MDCT), which has the advantage of time domain alias cancellation property, good de-correlation and

energy compaction. For psychoacoustic modeling and parametric modeling, they use DFT or dedicated filter banks. This will introduce mismatches and inefficiency in the case of low bit rate coding. If we can use the same time-frequency analysis block for adaptive quantization and psychoacoustic modeling, there will be advantages in performance of the coder.

5.Temporal Noise Shaping (TNS)

Temporal Noise Shaping is used to control the temporal shape of the quantization noise within each window of the transform. This is done by applying a filtering process to parts of the spectral data of each channel. The Pre-echoes is a problem with most block-based coding schemes. TNS uses frequency domain prediction to shape the quantization noise and make echoes, or noise, unnoticeable signal. It uses a filter to deal with original spectrum and quantizes. It transmits quantized filter coefficients to the bit-stream. The filter performed in the encoder, which leads to a temporally shaped distribution of quantization noise, and then the noise would not be noticeable in the decoded audio signal with implemented correctly [17].

Figure 3.5 shows the duality principle between time domain and frequency domain that if in the time domain there is a sample then, when it converted to frequency domain then it gives sine wave. So, if the sample is of noise then it will be spread out in whole frame. So, TNS will be used for limiting it. TNS gives the noise shaping which used with different orders.

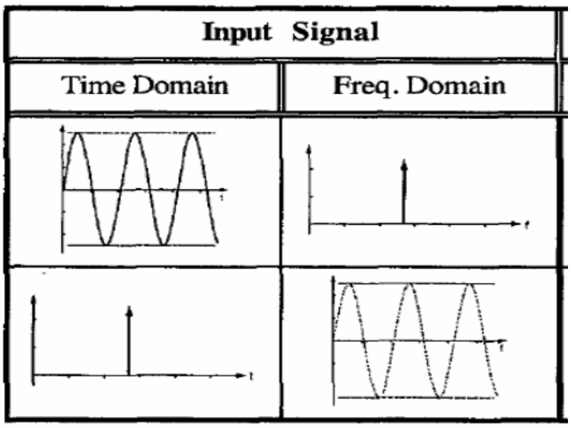

Figure 5 TNS processing encoder [8]

6.Quantization

The ultimate goal of quantization in AAC is twofold. The first goal, is the same as the quantization module for compression of any other multimedia signals, that is to reduce the necessary bits. The second goal, which is unique for audio coding, is to control the quantization noise. The quantization of AAC encoder exploits two nested loop, those are inner loop and outer loop to encode the spectral data and reduce redundent information [7] [8].

7.Source Coding

Here two coding techniques are introduced.

- Huffman Coding
- Run Length Coding

Huffman Coding Technique

Huffman coding is used to represent n-tuples of quantized coefficients, with the Huffman code drawn from one of 11 codebooks. The spectral coefficients within n-tuples are ordered and its size is two or four coefficients [2]. There are two codebooks for each maximum absolute value, with representing a distinct probability distribution function. The best fit is chosen. In order to save on codebook storage, most codebooks represent unsigned values. For these codebooks, the magnitude of the coefficients is coded (Huffman) and the sign bit of each non-zero coefficient is appended to the codeword. According to [2] the Huffman Coding is applied for coding.

Run Length Coding Technique

RLC stands for Run Length Coding. It is a lossless algorithm that only offers decent compression ratios in specific types of data. RLC is a very simple form of data compression in which *runs* of data (that is, sequences in which the same data value occurs in many consecutive data elements) are stored as a single data value and count, rather than as the original run. This is most useful on data that contains many such runs: for example, simple graphic images such as icons, line drawings, and animations. It is not useful with files that don't have many runs as it could greatly increase the file size.

RLE is probably the easiest compression algorithm. It replaces sequences of the same data values within a file by a count number and a single value. Suppose the following string of data (17 bytes) has to be compressed: *ABBBBBBBBBCDEEEEF* Using RLE compression, the compressed file takes up 10 bytes and could look like this: *A *8B C D *4E F* As you can see, repetitive strings of data are replaced by a control character (*) followed by the number of repeated characters and the repetitive character itself. The control character is not fixed; it can differ from implementation to implementation.

8.Bitstream formatting

Finally the Huffman/Run Length coded values are formed into a bitstream. A bit stream formatter is used to assemble the bitstream. The encoded bitstream consists of quantized and coded spectral coefficients along with some side information like bit allocation information, quantizer step size information etc.

III: MPEG-2 LC-AAC Decoder

The decoder of MPEG-2 LC-AAC is shown in Figure 3.6. From the Figure 3.6 one can see that the decoder has the entire reverse block than encoder. And finally the decoder will be implemented. Excluding the psychoacoustic block, all the other remaining blocks explain in chapter 4 will be implemented at decoder part and one can retrieve the original audio data at the decoder side.

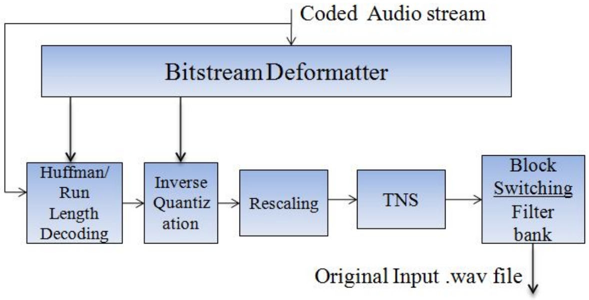

Figure 6: **Low Complexity Advanced Audio Decoder**

Principle of psychoacoustic Model (PAM)

The field of psychoacoustics has made significant progress toward characterizing human auditory perception and particularly the time-frequency analysis capabilities of the inner ear. Although applying perceptual rules to signal coding is not a new idea, most current audio coders achieve compression by exploiting the fact that "irrelevant" signal information is not detectable by even a well-trained or sensitive listener. Irrelevant information is identified during signal analysis by incorporating into the coder several psychoacoustic principles, including absolute hearing thresholds, critical band frequency analysis, simultaneous masking, the spread of masking along the basilar membrane, and temporal masking. Combining these psychoacoustic notions with basic properties of signal quantization has also led to the theory of perceptual entropy, a quantitative estimate of the fundamental limit of transparent audio signal compression.

IV:Psychoacoustic Model (PAM)

According to the ISO 13818-7 [2] standard, one can arrange the PAM into 13 steps. The block diagram of the PAM is shown in Figure 3.10 and the detail flow chart is shown as Figure 3.11. The 13 steps in calculating the masking threshold in the PAM as follows:

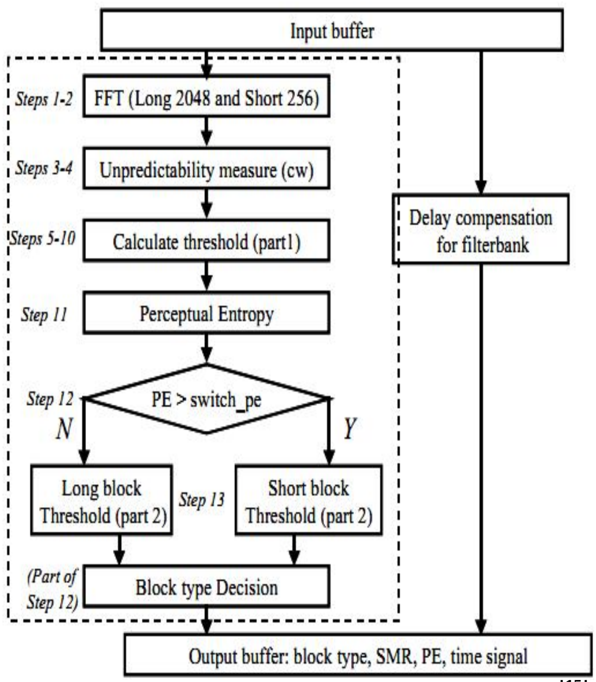

Figure 7 : **Block diagram of PAM (ISO 13818-7)** [15]

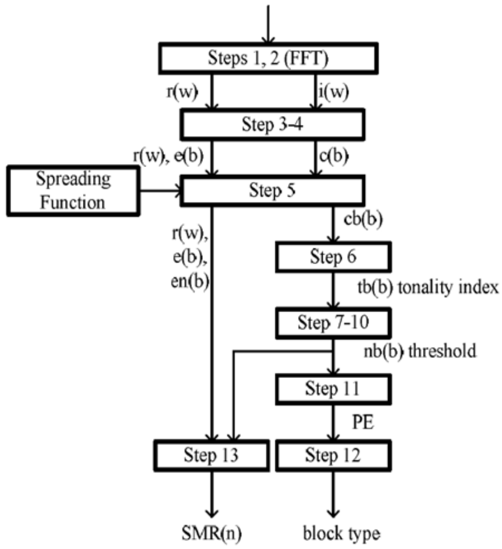

Figure 8: Detailed flowchart of PAM (ISO13818-7)[5]

V:SIMULATION RESULTS

Results from Psychoacoustic Model (PAM)

Input:

- ➢ Wave file: 'drumsA.wav'
- ➢ Sampling rate: 44.1 KHz
- ➢ Channel: Mono
- ➢ Coding: 16 bits PCM
- ➢ Bit rate: 705 Kbps
- ➢ Time duration: 1 second

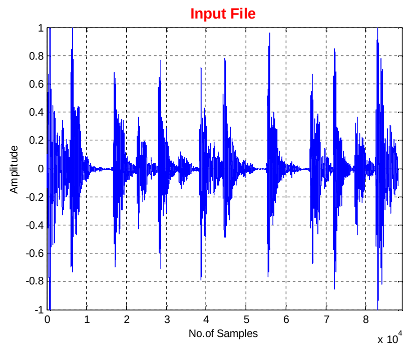

Figure 9: Input file: drumsA.wav

Output:

1.Set of Signal to Mask Ratio (SMR) and Signal to Noise Ratio (SNR)
2.Block type of MDCT (long, short, start or stop type)
3.Psychoacoustic Entropy (PE)
4.An addition to how many bits should be used for encoding in addition to the average available bits.

Graphs for various parameters of PAM (For frame No. 10):

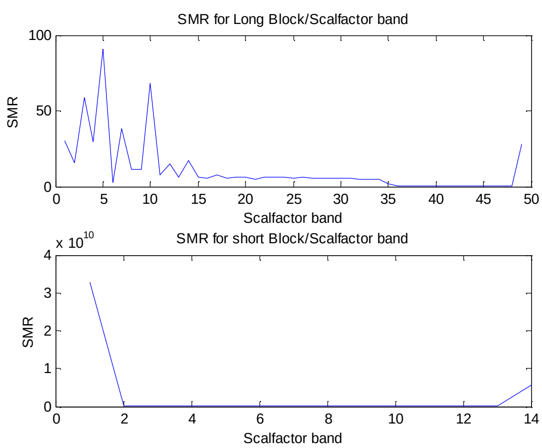

Figure 10: Plot of SMR per scalefactor band for long and short block

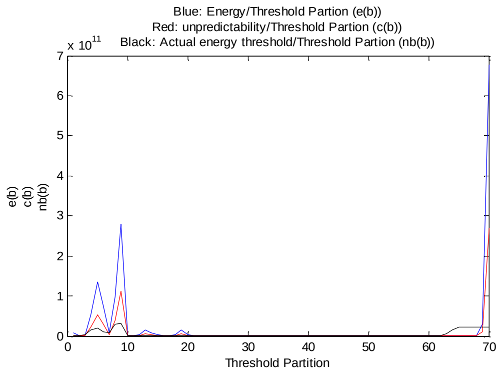

Figure 11: Plot of energy, unpredictability and actual threshold per threshold partition

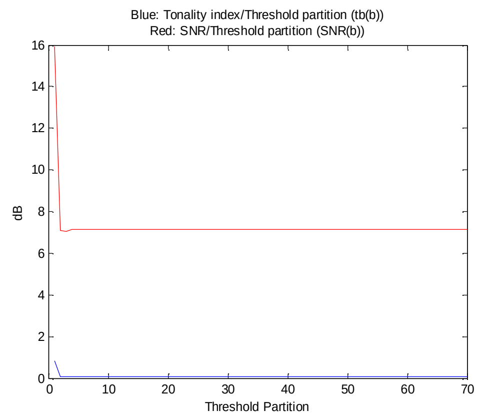

Figure 12: Plot of SNR per scalefactor band

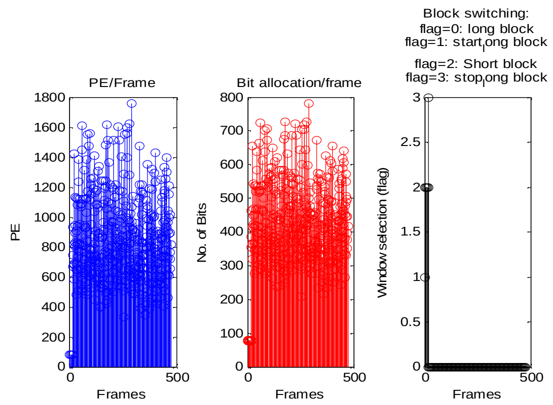

Figure 13: Plot of PE, bit allocation and bock switching decision per frame

Performance Measurement

As simulation results we perform two type of measurements

A. Objective analysis
B. Subjective analysis

A.Parameter for objective analysis

Here we analyze four objective parameters as follow:

- Signal to Noise Ration (SNR)
- Mean Square Error (MSE)
- Root Mean Square Error (RMSE)
- Analysis-by-synthesis Error (ABSE)

1.Signals-to-Noise Ratio

Signal-to-noise ratio (often abbreviated **SNR** or **S/N**) is a measure used in science and engineering that compares the level of a desired signal to the level of background noise. It is defined as the ratio of signal power to the noise power. A ratio higher than 1:1 indicates more signal than noise. While SNR is commonly quoted for electrical signals, it can be applied to any form of signal (such as speech, audio, video, image etc…).

Given the original speech x[n] and the synthetic version y[n], with the range of the time index n covering the measurement interval, the SNR is defined by,

$$SNR = 10\log_{10}\left(\frac{\sum_n x[n]^2}{\sum_n (x[n]-y[n]^2)}\right) \qquad (1)$$

2.Mean Squared Error (MSE)

For the original speech x[n] and the synthetic version y[n], with the range of the time index n covering the measurement interval, the MSE is defined by,

$$MSE = \frac{\sum_n (x[n]-y[n]^2)}{n} \qquad (2)$$

MSE shows the amount by which reconstructed speech differ from the original speech.

3.Analysis-by-synthesis (ABS)

It gives us the sample by sample difference between the original speech and recover speech.

$$ABS = sum\big(abs(x[n]-y[n])\big) \qquad (3)$$

B.Parameter for Subjective Analysis

Parameters like SNR (Signal to Noise Ratio) , ABSE (Absolute Error), MSE(Mean Square Error) are computed.

Parameters should be chosen so that their value is strongly correlated to the perceived quality loss. ITU provides a flawed and underspecified recommendation [4] to evaluate the perceived audio quality with two methods: Basic PEAQ (Perceptual Evaluation of Audio Quality) and Advanced PEAQ. Both are full reference quality indices: they are computed with respect to the original reference signal. The resulting indices are named as ODG (objective difference grade).

1.Objective Difference Grade

Usually audio, like voice, is also measured using MOS. ITU-R BS.1387 [10] defines Perceptual Audio Quality Measurement (PAQM) method. The basic idea of PAQM is very similar to PSQM although the internal algorithms are a bit different because of the inherent difference in audio and voice. The analysis of results from subjective listening test, in general, is based on Subjective Difference Grade (SDG) defined as: MOS signal under Test – MOS Reference. The Objective Difference Grade (ODG) is the output of PAQM. ODG corresponds to SDG in subjective domain.
The **objective difference grade** (ODG) is calculated by perceptual evaluation of the audio quality algorithm specified in ITU BS.1387-1 (PEAQ). It corresponds to the grade used in human-based audio tests. The ODG ranges from 0 to −4 and is defined as follows:

Impairment description	ITU-R Grade	ODG
Imperceptible	5.0	0.0
Perceptible, but not annoying	4.0	−1.0
Slightly annoying	3.0	−2.0
Annoying	2.0	−3.0
Very annoying	1.0	−4.0

Table 2: comparison table for ODG and ITU-R Grade (MOS)

Example :Simulation Results for file

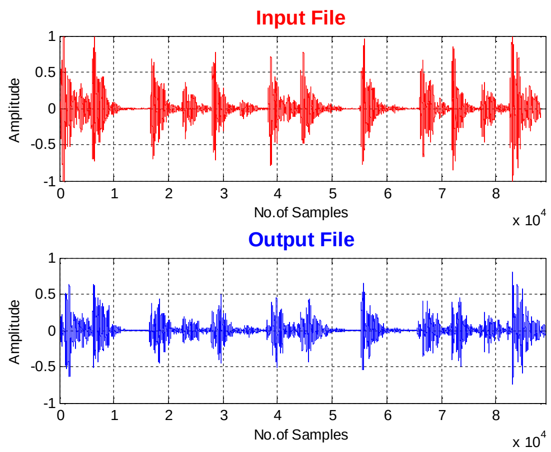

Figure 14: Input (Original) and Output (Decoded) waves; File: drumsA.wav

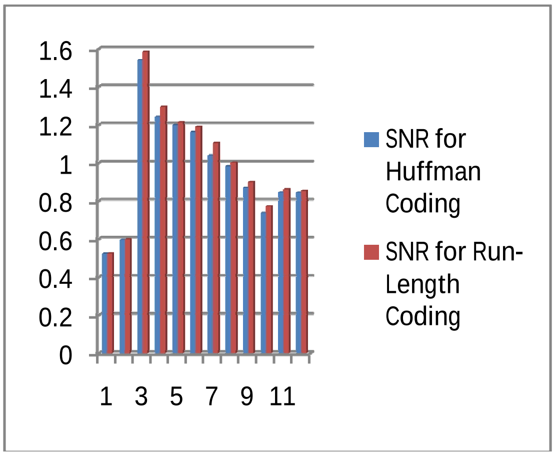

Figure 15: Comparison of SNR between Huffman Coding and Run Length Coding

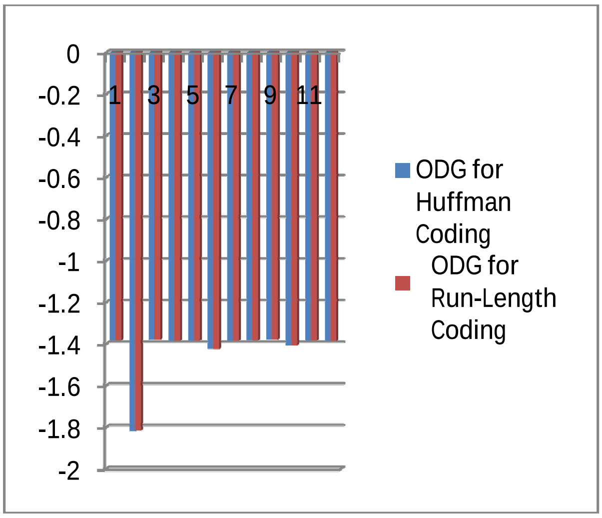

Figure 10: Comparison of ODG between Huffman Coding and Run Length Coding

CONCLUSION

The perceptual audio Codec functions very well in the MATLAB 9. The decoder can play bit-stream audio for 44.1 kHz sampling rate. The encoder can record the audio inputs from given wave file. The encoder and decoder can work perfectly at sampling rate 44.1 kHz. The audio qualities are quite good, we can see ODG is avg. -1.4 and MOS is avg. 3.6 (Table 6.2 and Table 6.3) at 44.1 kHz sampling rate and 64 Kbps bitrate.

From the Simulation results one can conclude that the psychoacoustic model is the crucial part of Low Complexity Advanced Audio Coder. The complexity of whole coder is depending upon the psychoacoustic model. So, when implementing AAC one can take care of computational complexity of psychoacoustic model. The LC-AAC is implemented with two different coding techniques that are using Huffman Coding and Run Length Coding.

But the noticeable thing is that, at the encoder side when an encoding is done with Huffman coding then one can get the bit-rate 64 kbps (Standard PCM bit-rate=705.5Kbps) means one can get 11(705.5Kbps/64Kbps=11) times compression. And when the Run Length Coding is used then it will be reach to 256 Kbps means there is a less compression (705.5Kbps/256Kbps=2.75) as compare with Huffman Coding. But at this bit-rate the performance parameters are same in both the techniques. So, Huffman coding technique generally is used as source coding in AAC, all though by using the Run Length Coding technique one can get satisfactory results.

REFRENCES

[1] Peter Noll, "MPEG Digital Audio Coding," *IEEE Signal Processing Magazine*, pp. 59-81, September 1997.

[2] J. D. Johnston and A. J. Ferreira, "Sum-difference stereo transform coding," *IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 2, pp. 569–572, Mar. 1992.

[3] H. Fuchs, "Improving joint stereo audio coding by adaptive inter-channel prediction," *IEEE Conference on Applications of Signal Processing to Audio and Acoustics*, pp. 39–42, Oct. 1993.

[4] A. Jin, T. Moriya, T, Norimam, M. Tsushima and T. Ishikawa, "Scalable Audio Coder Based on Quantizer Units of MDCT Coefficients," *IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 2, pp. 897-900, Mar. 1999.

[5] S. Kuo and J. D. Johnston, "A study of why cross channel prediction is not applicable to perceptual audio coding," *IEEE Signal Processing Letters*, vol. 8, No. 9, pp.245-247, Sept. 2001.

[6] Mark Kahrs and Karlheinz Brandenburg, "Applications Of Digital Signal Processing To Audio And Acoustics," Kluwer Academic Publishers, 2002.

[7] Han-gil Moon, Jeong-il Seo, Seungkwon Baek, and Koeng-Mo Sung, "A Multi-Channel Audio Compression Method with Virtual Source Location Information for MPEG-4 SAC," *IEEE Transactions on Consumer Electronics*, Vol. 51, No. 4, pp.1253-1259, Nov. 2005.

[8] Andreas Spanias, Ted Painter and Venkatraman Atti, "Audio signal processing", "A John Wiley & Sons, Inc., Publication", 2007.