IMPLEMENTATION AND OPTIMIZATION OF FIXED POINT MP3 DEOCDER TARGETED FOR ARM7TDMI

Thejus Nambiyar
Sathyabama Institute of Science and Technology, Chennai, India.

Abstract

Nowadays, most common MP3 decoders in market are designed for high end Advanced Risc Machine (ARM) chips. The biggest drawback for the same is that, they are costly, as well as complex. Besides these factors, the time consumption is high for the current decoders since the million code instruction per second(MIPS) is high. Considering these drawbacks, the feasibility for the current mode of decoders is less. In this paper, we put forward the idea to implement an MP3 decoder that is optimized for ARM7TDMI chip. The advantage of this is that it is a low cost chip, and the power consumption is less, hence reducing the overall cost. The main objective of this decoder is to convert the MP3 file into PCM format, that is, Pulse Code Modulation, which is the raw layout of sound. Also, the size of this file will be large.

Keywords—MIPS, PCM Format, ARM chips, MP3 Decoders, ARM7TDMI chips.

I.INTRODUCTION

Audio Codec is generally termed for a device which can code or decode[1] a digital data stream of audio. In terms of software, an Audio Codec is a computer program that compresses or decompresses Digital audio data according to a given audio file format or streaming media audio format. This program implements an Algorithm for carrying out the process. Mostly, the audio formats are of MP3 type, and it is being decoded to PCM format, which is the raw format for an audio.

The current MP3 fixed point decoders has its own pros and cons. One of the major shortcomings for these decoders is their cost[2]. The decoder complexity also adds to their drawbacks. Also the high time consumption is another constraint. The reason behind these decoders being highly complex and sophisticated is because they are designed for high end ARM chips.

This project proposes the implementation of Fixed point MP3 decoder for the same ARM processor from C language[3]. Also, the million instructions per second will be improved by optimizing the above C code to Assembly language for ARM processor. This will hence increase the through put for the entire decoder. Also the efficiency will be more when compared to the traditional decoders. Also, it is to be mentioned that the chip used here has very low power consumption.

The algorithm being implemented for the above mentioned functionalities is the Huffman Decoding Algorithm[4]. This algorithm focuses on compression and decoding of a given set of information/data. The compression and decoding is done with the help of variable-length coding table. The estimated occurrence likelihood of each possible value of the source symbol is calculated. Based on this estimated calculation, the variable-length code table is derived. The representation of Huffman Coding Table (HCT) and the related decoding algorithm is ambiguous. The most common or traditional way of representing the HCT is using the Binary Tree Approach, that is, the Huffman Tree as it is being called. The binary search techniques are used on this for decoding process. Nevertheless, using Binary Tree for HCT representation does not seem feasible in real time application due to time and memory constraints. Some of the other efficient ways of representing Huffman Table is by using Arrays. The Huffman tables will be optimized by representing it as an array, and also the efficiency will be increased by reducing the time and memory constraints.

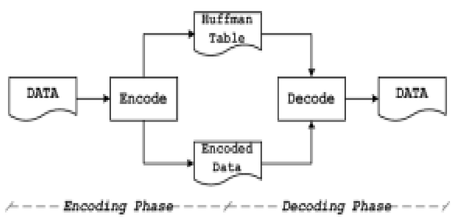

Fig.1. Huffman coding technique diagaram

II.SOFTWARE AND HARDWARES USED

The ARM7TDMI is used in this work. The software that helps producing sound is being fed into this chip. In this process, the MP3 format will be decoded into the raw format of sound, PCM.

Following are the software being used in this project work. The code is developed in MS Visual Studio Platform which is used to debug the C++ code, and in addition to that, Adobe Audition, ARM Debugger, ARM Developer Suit are also being used. The functionalities of each of the software vary in different aspect. For example, the sound in PCM format can be heard using Adobe Audition. Likewise, the code that is to be written for ARM chip is debugged using ARM debugger, whereas the debugged code is being fed to the chip is done using ARM Developer Suite.

III.DESIGN

The Pictorial representation or the block chart of the system architecture can be seen below:

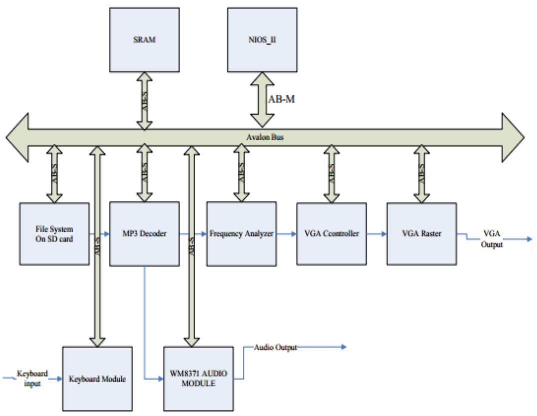

Fig.2. System architecture

IV.DESCRIPTION OF MODULES

The first module of this part of work is the header File Generation. Here, there are a couple of header files, such as assembly.h, coder.h, debug.h, etc.

Secondly, the initialization of Timer and Debug of Memory Check is carried. This is the second module. Here, at the beginning of the method, timer is initialized as zero. The change in timer will start once the entire process starts. It should be noted that, at the initial stage of the process, memory check should be debugged.

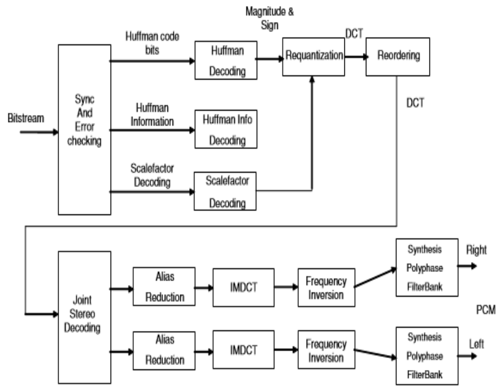

Fig.3. Data flow diagram

Decoder technique is the next module description. The prime focus of this module is to convert the MP3 into the raw format of sound, which is PCM. PCM format helps in producing the sounds from an amplifier.

The final module comprises of output file generation, which is of PCM format. The PCM format which is the final output file is read using Adobe audition software.

V.TESTING METHODOLOGIES

The testing methodologies adopted here are mostly the conventional Unit testing, System and Integration Testing. We implement these traditional testing strategies in various other tests that is relevant to our Project work.

The following tests are carried out in this project:
-- Full Decode
-- VBR Compatibility
-- 8191+ / 100Hz bug
-- Objective Sound Quality
-- 16 Bit Least significant Bit Accuracy
-- 24 Bit Accuracy
-- ID3v2 Accuracy
-- Corrupt File Recovery
-- Free Format compatibility
-- Codec Listening Test
-- ABX Test
-- ABC/HR Test
-- MUSHRA Test

Each of these tests carries out different functionalities to test some particular features. The tests mentioned above contain unique testing techniques to test the module functionalities, ensuring a bug free end product.

VI.EXPERIMENTS CONDUCTED

The input used in this project is a complete MP3 stream. An Mp3 stream structured in frames of bits, where each frame consists of 1152 encoded PCM samples. The frame size within a single stream may not be constant. They can vary. Hence, the information about the frame detection should be contained in the frame header. The logical sections of encoded frame can be depicted with the help of the below pictorial representation.

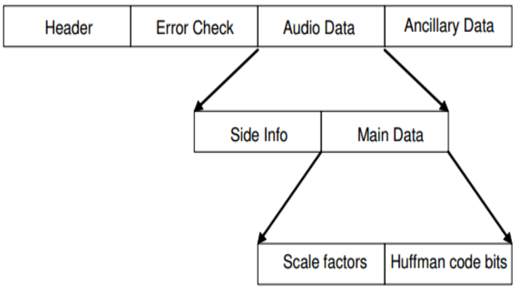

Fig.4. MPEG 1 Layer 3 frame format

The main data used here consists of coded scale factors and Huffman Coded bits. The scale factors referred here is used in decoder in order to get division factors for a given group of values. They are termed as scale factor brands. The selection for these groups is based on various frequencies and non uniform response of human ear. Huffman codes are used for encoding these quantized values.

The Ancillary data field is the private data, where the extra information like ID3, and artist information and name of the songs are sent by the encoder.

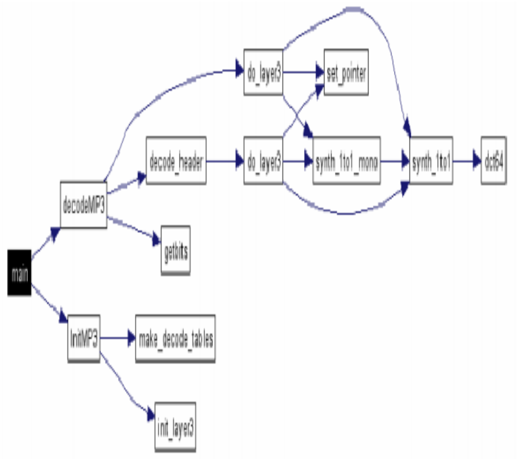

Fig.5. Call graph of major functions in the reference C implementation

VII.CONCLUSION

The prime motive of this project is to decrease the overall cost of the chip when compared to other chips in place. Also, for the ARM7TDMI chip, the code is also optimized and hence, the time and other constrains are also eliminated. Another highlighting advantage for these chips is that consumption of power is very insignificant when compared to other chips like ARM9TDMI. In addition to the optimized software, the million instructions per second for this chip has been decreased, which will result in a comparatively negligible time consumption.

REFERENCES

[1]. Jean-Marc Valin , Gregory Maxwell, Timothy B. Terriberry, and Koen Vos,"High-Quality, Low-Delay Music Coding in the Opus Codec", 135th AES Convention,Xiph.Org Foundation The Opus Codec, 2013, October 17–20 New York, USA.

[2]. Keun-Sup Lee, Young Cheol Park and Dae Hee Youn," SOFTWARE OPTIMIZATION OF THE MPEG-AUDIO DECODER USING A 32-BIT MCU RISC PROCESSOR", Dept. of Division of Information Technology, Yonsei University, Wonju, Korea

[3]. Koen Vos, Karsten Vandborg Srensen, Sren Skak Jensen and Jean-Marc Valin,"Voice Coding with Opus", 135th AES Convention, The Opus Codec, 2013 October 17-20 New York, USA

[4]. Vijaykuinar Gurkhe," OPTIMIZATION OF AN MP3 DECODER ON THE ARM PROCESSOR" TENCON 2003