

# Efficient Implementation of Ogg Vorbis Decoder Using Soft Core Processor

Sapna Pawar<sup>1</sup>, Jitendra Zalke<sup>2</sup>

<sup>1</sup> (M.Tech 4th Sem, Department of Electronics Engineering,  
Shri Ramdeo Baba College of Engg & Management, Nagpur

<sup>2</sup> (Assistant Professor, Department of Electronics Engineering,  
Shri Ramdeo Baba College of Engg & Management, Nagpur

## ABSTRACT

This paper describes a design of Ogg vorbis decoder for an embedded platform. Ogg Vorbis is an audio compression format developed by the Xiph.Org Foundation, a non-profit organization working in the area of Internet multimedia technology. Since ogg vorbis decoding process incurs high computational complexity, a software based implementation requires high operational frequency. Thus in our design a hardware module will be designed for high computational complex block. We aimed to implement this design using soft core processor MicroBlaze by xilinx.

**Keywords:** ogg, vorbis, tremor, microblaze processor, imdct.

## 1. INTRODUCTION

With recent advances of the audio compression technology, the variety of net-based digital music services, such as internet Broadcasting and online music stores are attaining greater popularity. The most renowned compression technique in use today is MPEG I layer III, or "MP3", which is even used as a synonym for compressed music data. Even though MP3 has gained wide spread acceptance, it suffers from various flaws and limitations. One notable limitation is that MP3 does not support more than two audio channels.

The development of Ogg Vorbis started as a reaction to the license policy run by the Fraunhofer institute, owner of the MP3 standard. Ogg Vorbis [1] is currently a growing audio format, mainly used for online distribution of music. It is totally license and royalty free. For embedded platforms the licensing and royalty cost for supporting commercial formats can be quite taxing as payments are often per device.

The result of [2] comparison of audio compression format demonstrates the superiority of ogg vorbis in audio quality as well as it supports multichannel. However computational cost of ogg vorbis decoding process is higher than that of mp3. Therefore in portable audio appliance which can perform both mp3 & ogg vorbis software decoding, an embedded processor must run at high operation frequency to decode the ogg vorbis stream. These technical issues motivated us to design an ogg vorbis decoder for embedded system.

## 1. OGG VORBIS

### 2.1 INTRODUCTION

Ogg Vorbis is an audio compression format developed by the Xiph.Org Foundation, a non-profit organization working in the area of Internet multimedia technology. Ogg itself is a big framework for several multimedia projects including Vorbis (audio) and Tarkin (video).

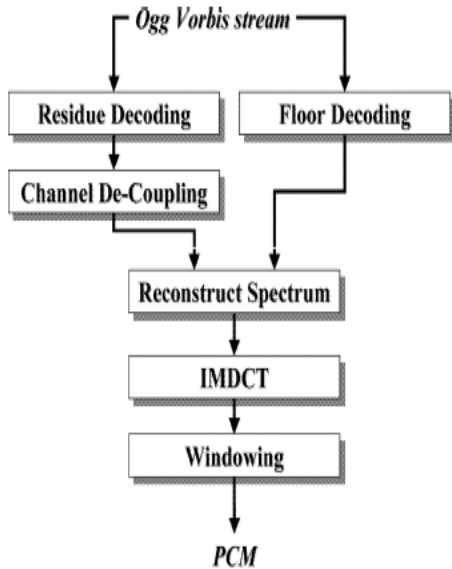
Vorbis is the first project in the Ogg family aimed at the audio compression/decompression.

Encoding and decoding of Ogg Vorbis can be classified in 6 big stages. Vorbis encoder takes the raw audio data as overlapped but contiguous short-time segments and analyses the audio data to find the optimal small representation. This stage is called analysis. After that it encodes the audio data into a much smaller data representation as determined in the previous step. This stage is called coding. Then the raw packets will be packed into streams, called streaming.

At the other end, the decoder extracts the sequence of raw packets from the stream; the stage is called streaming-decomposition. It then tries to reconstruct the sound signal representation from these packets, called decoding. Lastly, the audio signal will be regenerated from the decoded representation in the synthesis stage.

### 2.2 DECODING PROCESS

In our project, only decoder part is of interest (see fig 1). Decoding is straight forward and less complex than encoding. Decoder decomposes the Vorbis packet out of the Ogg stream. Vorbis packet is then processed in the Decoding stage to extract floor, residue and do channel-decoupling. The audio signal in frequency domain is recovered, inverse-MDCT transformed back to time domain and de-overlapped to form the output audio signal.



**Fig 1 Ogg vorbis decoding process**

**2.3 RELATED WORK**

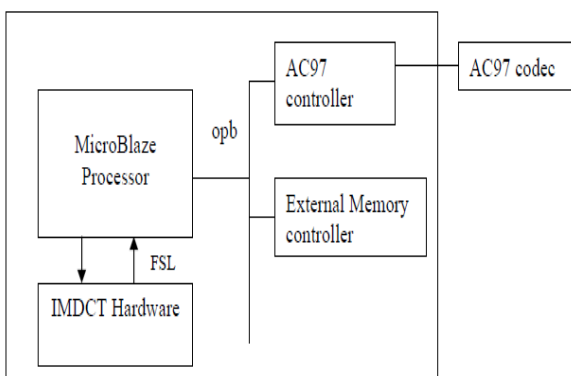
During last few years, the ogg vorbis decoder is designed & implemented as an embedded system for different soft core processors.

Ogg Vorbis Decoder For Motorola DSP56002 [3], Ogg/Vorbis in embedded systems [4] for DSP Processor & Design of an Audio Player as System-on-a-Chip [5] for LEON SoC Platform.

In this paper the ogg vorbis decoder with specific hardware using microblaze processor will be implemented.

**2. DECODER ARCHITECTURE**

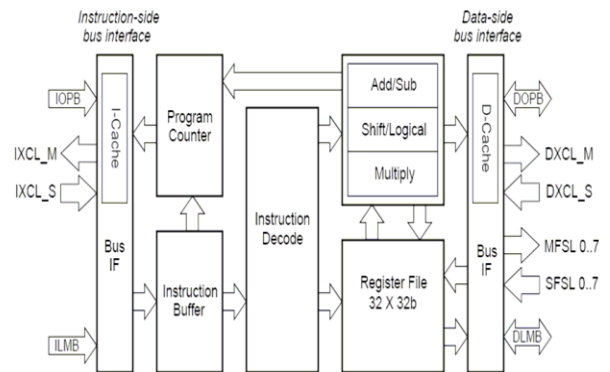
The fig 2 shows the architecture of our proposed ogg vorbis decoder which mainly consists of Microblaze processor, AC97 sound device controller & decoder part of ogg vorbis implemented as an IMDCT hardware. A fixed-point decoder called Tremor [6] (developed by Xiph.Org the creator of the Vorbis I specification), will be ported to the MicroBlaze processor and an Ogg Vorbis player will be developed. IMDCT hardware is explained next in section 3.2



**Fig 2 Decoder Architecture**

**3.1 MICROBLAZE PROCESSOR**

The MicroBlaze embedded soft core [7] is a reduced instruction set computer (RISC) optimized for implementation in Xilinx field programmable gate arrays (FPGAs).



**Fig 3 Microblaze core block diagram**

The MicroBlaze embedded soft core is highly configurable, allowing users to select a specific set of features required by their design.

**3.2 IMDCT (INVERSE MODIFIED DISCRETE COSINE TRANSFORM) HARDWARE**

MDCT is linear orthogonal lapped transform which is introduced by princeton in [8]. up to now MDCT is used widely in the state-of-the-art-audio codecs such as MP3, AAC, Ogg vorbis

Let  $x(k)$  be the samples in the time domain and  $n$  the size of the block.  $x_t(k)$ ,  $k = 0..n-1$  are the samples used to calculate the frequency domain samples  $X_t(k)$ ,  $k = 0..n/2-1$  of the block number  $t$ .

The equation of the direct MDCT is:

$$X_t(m) = \sum_{k=0}^{n-1} f(k)x_t(k) \cos\left(\frac{\pi}{2n}(2k+1+\frac{n}{2})(2m+1)\right)$$

*for  $m = 0..n/2-1$*

The equation of the inverse MDCT is:

$$y_t(p) = f(p) \frac{n}{4} \sum_{m=0}^{n/2-1} X_t(m) \cos\left(\frac{\pi}{2n}(2p+1+\frac{n}{2})(2m+1)\right)$$

*for  $p = 0..n-1$*

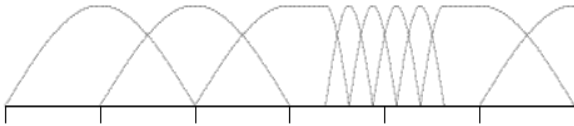
Cancellation of time domain alias terms is done by an overlap add operation:

$$\tilde{x}_t(q) = y_{t-1}\left(q+\frac{n}{2}\right) + y_t(q) \quad \text{for } q = 0..n/2-1$$

To cancel alias terms the shape of each window must keep the following conditions: The shapes of the windows in succeeding blocks must fit to each other only in the overlapping part. It is possible to split each long block into shorter blocks. Overlapping these shorter blocks must

result in the same window shape as used by the overlapping part of a long block.

$$f_{r-1}\left(\frac{n}{2} + k\right)^2 + f_r(k)^2 = 1 \quad \text{for } k = 0..n/2-1$$



**Fig 4 typical succession of window shapes**

In ogg vorbis codec the window function is given by:

$$f_t = \sin(2\pi \sin^2(t/n)).$$

### 3. CONCLUSION

This paper has described the process of efficient implementation of ogg vorbis decoder as an embedded system. Since the software only implementation of decoder will result in slow speed, so to increase the computational speed hardware module will be designed. The hardware module for IMDCT of ogg vorbis decoding process will increase the computational speed.

### 4. REFERENCES

- [1] "The ogg vorbis codec project", <http://www.xiph.org/vorbis/>
- [2][http://en.wikipedia.org/wiki/Comparison\\_of\\_audio\\_for\\_mats](http://en.wikipedia.org/wiki/Comparison_of_audio_for_mats)
- [3] Niklas Barsk, Ogg Vorbis Decoder For Motorola DSP56002, Linköpings Tekniska Högskola, 2005
- [4] Erik Montnémy, Johannes Sandvall, Ogg/Vorbis in embedded systems, Lunds Tekniska Högskola, 2004
- [5] Luis Azuara, Pattara Kiatisevi, "Design of an Audio Player as System-on-a-Chip", University of Stuttgart, 2002.
- [6] <http://xiph.org/vorbis/tremor>
- [7] MicroBlaze Processor Reference Guide Embedded Development Kit EDK 13.1, [www.xilinx.com](http://www.xilinx.com)
- [8] B. Edler, Kh. Brandenburg, Th. Sporer, "The use of multirate filter banks for coding of high quality digital audio", 6th European Signal Processing Conference (EUSIPCO) (1992).
- [9] EDK Concepts, Tools, and Techniques, A Hands-On Guide to effective Embedded System Design EDK 13.1UG683 April 13, 2011, [www.xilinx.com](http://www.xilinx.com)

- [10] Kosaka, A.; Yamaguchi, S.; Okuhata, H.; Onoye, T.; Shirakawa, I "Design of ogg vorbis decoder system for embedded platform" IEICE TRAN. FUNDAMENTALS, VOL.E88-A, NO.8 AUGUST 2005.