A Hardware Implementation of Ogg Vorbis Audio Decoder with Embedded Processor

Atsushi Kosaka[1], Satoshi Yamaguchi[1], Hiroyuki Okuhata[2],
Takao Onoye[1] and Isao Shirakawa[1]
[1]Dept. Inf. Sys. Eng. Osaka University
2-1 Yamada-Oka, Suita,
Osaka, 562-0036 Japan
Phone: +81-6-6879-7808, Fax: +81-6-6875-5902
Email : kosaka@ise.eng.osaka-u.ac.jp, {sato, onoye, shirakawa}@ist.osaka-u.ac.jp
[2]Synthesis Corporation
2-1-11 Senba-Nishi, Mino,
Osaka, 565-0871 Japan
Phone: +81-727-27-8162, Fax: +81-727-27-8163, Email : okuhata@synthesis.co.jp

Abstract:

A VLSI architecture of an Ogg Vorbis decoder is proposed, which is dedicated to portable audio appliances. Referring to the computational cost analysis of the decoding processes, the LSP (Line Spectrum Pair) process, which takes more than 50% of the total processing time, can be regarded as a bottleneck to achieve realtime processing by embedded processors. Thus in our decoder a specific hardware architecture is devised for the LSP process so as to be integrated into a single chip together with an ARM7TDMI processor. In addition, in order to reduce the total hardware cost, instead of the floating point arithmetic, the fixed point arithmetic is adopted.

The LSP module has been implemented with 9,740 gates by using a Virtual Silicon 0.15μm CMOS technology, which operates at 58.8MHz with the total CPU load reduced by 57%. It is also verified that the use of the fixed point arithmetic does not incur any significant sound distortion.

1. Introduction

With the recent progress of network technologies and audio compression algorithms, a variety of *net*-based digital music services, such as Internet broadcasting and online music stores, are attaing great popularity.

Up to now, the most widely used audio compression algorithm is MP3 (MPEG Audio Layer-III)[1]. However, the licensing royalty on patents of the audio compression technologies are now recognized as inevitable issues, and hence MP3, on which a royalty of distributing encoders, decoders, and encoded data files is charged, is going out of the core technologies in the network distribution business.

On the contrary, Icast Co., Ltd. is developing a novel audio compression algorithm, called Ogg Vorbis[2], which is distinctive in terms of fully open, non-proprietary, and patent-and-royalty-free distribution. Thus, from the viewpoint of these *ease-to-use* features, Ogg Vorbis can be regarded as a promising next-generation core technology in the network distribution business.

The realtime decoding by Ogg Vorbis can be achieved with the use of a high performance CPU such as a PentiumIII processor. However, considering that the use of Ogg Vorbis in portable applications suffers from limited memory size and high power consumption, there still remains much room to develop an efficient VLSI architecture dedicated to portable embedded applications.

Motivated by this technical issue, the present paper constructs an embedded processor based architecture for the Ogg Vorbis decoding. According to the computational cost analysis of an ARM7TDMI processor with the use of sample coded data files, Reconstruct Curve process takes more than 55% of the total processing time, failing in the realtime processing. Since the LSP [3][4] calculation dominates this Reconstruct Curve process, a specific functional module should be devised for LSP. Apart from this LSP process, other processes should be run on an ARM7TDMI processor in order to reduce the total hardware cost.

The rest of this paper is organized as follows. Section 2 summarizes an Ogg Vorbis algorithm, Section 3 describes the computational cost analysis by using an ARM7TDMI processor, Section 4 details the proposed Ogg Vorbis decoder architecture, Section 5 shows implementation results and audio quality assessment using the proposed algorithm, and finally, Section 6 addresses concluding remarks.

2. Ogg Vorbis Algorithm

2.1 Overview

In the same manner as other audio coding algorithms, the Ogg Vorbis encoding is performed frame by frame, which is composed of 1,024 (Long frame) or 128 (Short frame) samples of audio input data. Fig. 1 shows a general flow of the algorithm. Each encoding module is summarized as follows.

- MDCT (Modified Discrete Cosine Transform) and FFT (Fast Fourier Transform)
  Coefficients in the time domain are transformed to those in the frequency domain by MDCT and FFT. In the succeeding process, FFT coefficients are used for the tonal estimation, and MDCT coefficients for the noise analysis.

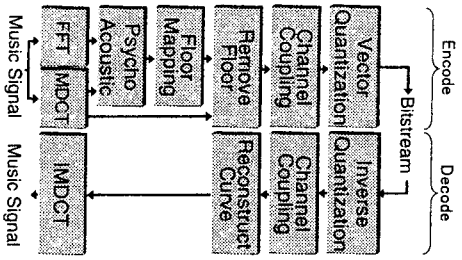

Figure 1. Process flow of Ogg Vorbis

- Psycho-Acoustic Model
  In this process, inaudible spectrums are removed on the basis of Psycho-Acoustic Model so as to reduce the data amount without degrading the audio quality.
- Floor Mapping
  Floor Mapping is to generate the spectrum envelope, i.e. floor curve, which represents the change of spectrums in the frequency domain by using LPC (Linear Predictive Coefficient). LPC is calculated by the linear predictive analysis of spectrums.
- Remove Floor
  In this process, the MDCT coefficients are divided by their floor curve values to flatten the shape of spectrums as well as to reduce the quantization errors. Details of this process are described in the next section.
- Channel Coupling
  Channel Coupling is to transform the data of the left and right channels into the magnitude and angle components, reducing the data amount through the use of the correlation of two channels.
- Vector Quantization
  Vector Quantization is to treat spectrums not individually but collectively. A set of spectrums which represents a particular region, is approximated by a codevector. The set of all codevectors is called the Codebook.

To the contrary, the decoding algorithm is accomplished by tracing above mentioned processes in the reverse order, as indicated in Fig. 1.

2.2 Remove Floor and Reconstruct Curve

Remove Floor is a functional block of the encoding process, in which each frequency spectrum is divided by the corresponding value of the spectrum envelope, called the floor curve, which has a characteristic similar to human psycho-acoustic. This operation flattens the shape of frequency spectrums and reduces the quantization noises. Subsequently, the flattened frequency spectrums are converted into the Bark scale[5], an exponential scale.

Reconstruct Curve in the decoding is to reconstruct the floor curve by transforming spectrums in the Bark scale into those in the original scale, in order to generate the spectrum envelope. Finally, the frequency spectrums, which represent audio data, are reconstructed by multiplying spectrum envelope values.

In order to determine the floor curve, first LPCs (Linear Predictive Coefficients) are calculated by the linear predictive analysis of spectrums. Then LPCs are quantized after they are converted to the LSP form, which represents the floor curve using pairs of resonance frequencies, in order to attain high quantization efficiency.

3. Computational Costs Analysis

In order to construct efficient architecture of an Ogg Vorbis decoder for embedded applications, which has different limitations in terms of hardware cost and power consumption, the computational cost for each functional block in the decoding process is analyzed by software simulation, where an ARM7TDMI processor is used as the target processor, since it of wide commercial use.

3.1 Software Simulation

The Ogg Vorbis decoding process is analyzed by ARMulator (an emulator of ARM processors). Samples bitstreams used in this analysis are *violin*, *harp*, *pops* and *voice* (voice of a man) encoded by the Ogg Vorbis encoder at 44.1kHz sampling rate with max bitrate 128kbps. Table 1 summarizes the number of cycles per frame in decoding process.

Table 1. Analysis of decoding process

	play time [sec]	frame	cycles/frame
violin	10	638	3,145,027
harp	10	855	4,344,035
pops	16	1,503	3,810,080
voice	20	2,249	3,386,672
worst case	-	-	4,344,035

Among these sample streams, *harp* is in the worst case, for which the operation frequency is sought by using the cycles/frame. The time allocated for 1 frame (1,024 samples) is given by

$$1024 \times \frac{1}{44,100} = 23.2 \text{ [ms]}. \quad (1)$$

In order to fulfill the realtime processing requirement, the processor has to perform all the operations within the allocated time, that is, the processor must run at the operation frequency of

$$\frac{4,344,035}{0.0232} = 187.242 \text{ [MHz]}. \quad (2)$$

It can be easily seen that ARM7TDMI can hardly realize the requirement, and hence a specific hardware is employed to enable the realtime decoding. However, the employment of hardwares for all decoding processes may suffer from an impractical hardware cost for mobile applications.

Considering these situations, as an Ogg Vorbis decoder architecture a functional block with higher computational complexity should be implemented by a specific hardware, with the others performed by CPU, so as to reduce the total circuit size and power consumption.

3.2 Module Profiling

To create the profile summary of decoding process, each sample (*violin*, *harp*, *pops*, *voice*) has been decoded three times by using ARMulator, to attain the average of three profile data for each sample. Table 2 summarizes the results of module profiling.

Table 2. Profile summary - ARM7TDMI -

	Reconstruct Curve	IMDCT	IQ	etc
violin	57.3%	28.9%	2.4%	11.4%
harp	56.6%	30.2%	2.5%	10.7%
pops	55.2%	27.0%	3.1%	14.7%
speech	59.1%	28.0%	2.4%	10.5%
average	57.1%	28.5%	2.6%	11.8%

From this table it can be seen that Reconstruct Curve and IMDCT have higher computational complexity than the other blocks. Specifically, the computation load of the LSP process in Reconstruct Curve amounts more than half of the total.

As a result, it turns out that employing a specific hardware for LSP can reduce the required operational frequency of the embedded processor up to the practical value, suppressing the hardware overhead.

4. Decoder Architecture

4.1 Overall Organization

Fig. 2 shows an overall organization of the proposed Ogg Vorbis decoder, which consists mainly of embedded processor, LSP functional module, Internal Memory, I/O interface for external Memory, and peripherals.

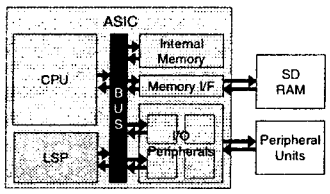

Figure 2. Block diagram of envisaged system

In the process of the Ogg Vorbis decoding, the processor stops the decoding when the LSP process is invoked. On receipt of the start signal from the processor, the LSP process starts, and at the termination the LSP process writes out the result in the memory to generate an interrupt to the processor.

4.2 LSP Module

In the LSP process, the cosine and exponent operations are frequently invoked, and these operations are to be performed by the table look-up and the linear interpolation so as to attain small circuit size and low operation complexity.

Cosine operation can be approximated by

$$\cos x \simeq T_{x'} + \alpha(T_{x'+2\pi/N} - T_{x'}) \times x'',$$

$$x' = m \times \frac{2\pi}{N},\ x'' = x - x' \qquad (m = 0, 1, \cdots, N-1),$$

where T_i, N, and α represent the value taken from the table, the number of elements of table, and constant, respectively. Fig. 3 shows an architecture of Cos Operation, which performs an approximated cosine operation.

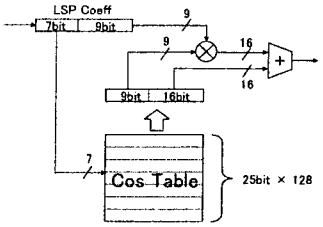

Figure 3. Cos Operation

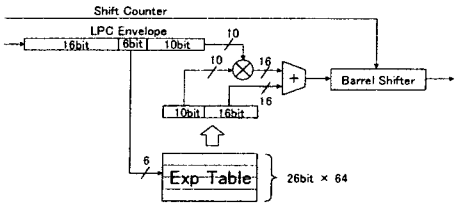

Figure 4. Exp Operation

In this architecture, a sufficient operation accuracy is attained with 128 table elements. Each table element consists of a 16-bit cosine value and a 9-bit difference between the present and next elements, by which a cosine operation can be performed in a single cycle without referring to multiple table entries.

As mentioned before, the spectrum envelope is expressed with the exponential scale, called Bark scale, and therefore the LSP process transforms a spectrum envelope into that of the original scale. Exp Operation performs this transformation, which is implemented by the same architecture as Cos Operation, where the table consists of 26-bit×64 elements, as illustrated in Fig. 4.

Fig. 5 shows a detailed structure of the LSP module based on the proposed architecture.

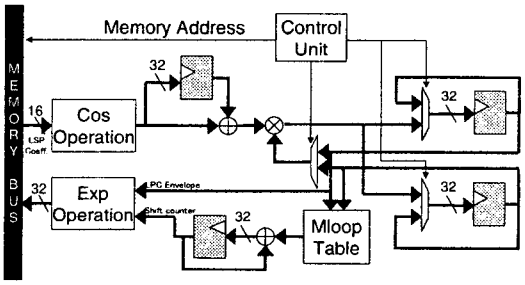

Figure 5. LSP hardware

Mloop Table is the table to determine the shift for the register value, which is held by another register. Control

Unit controls the state of the LSP process, and memory access.

Since the employment of the floating point operation increases the circuits size, the LSP module is implemented by using the 32-bit fixed-point arithmetic, in which the upper 16-bit represents integer part and the lower 16-bit the decimal part.

In the LSP process, the multiplication is repeatedly applied to the LSP coefficients. Since a great number of multiplications are necessary in the LSP process, the multiplications may affect the total computational complexity of the LSP process. Therefore one 32-bit multiplier which operates in a single cycle is implemented in the LSP module.

5. Implementation Result and Quality Assessment

5.1 Audio Quality Assessment

In order to check the audio quality, ogg streams using the fixed-point arithmetic LSP are compared to those using the floating point arithmetic LSP. SA (Subjective Assessment), OA (Objective Assessment) and SNR (Signal to Noise Ratio) are adopted as the measures of assessment, where SA is based on 5-class evaluation by several peoples, and OA is based on ODG, which is the audio quality measurement on a five-grade scale from -4 (very annoying disturbance) to 0 (imperceptible difference), based on ITU-R BS.1387[6].

Table 3 summarizes the results of SA, OA and SNR of 4 samples (*voice* - voice of a man, *flute* - sound of a flute, *classic* - classic music, *rock* - rock music) coded at 128kbps.

Table 3. Audio quality assessment.

	SA	OA	SNR [dB]
voice	4.875	−0.0748	28.718
flute	5.000	−0.0185	17.750
classic	5.000	−0.1029	32.882
rock	4.500	−0.1111	29.475

Since fixed-point values are rounded, all the SN ratios are less than 33 [dB]. However, the results of SA is almost 5, and those of OA almost 0, indicating that the audio quality degradation by implementing LSP using the fixed-point arithmetic is imperceptible by the human ear.

5.2 Implementation Result

The LSP module based on the proposed architecture has been implemented through the use of Synopsys *Design Compiler* by the Virtual Silicon 0.15μm CMOS technology. Table 4 summarizes the main features of the LSP module.

Table 4. Main features of LSP unit.

0.15 μm CMOS technology	
Number of Gates	9,740
Max. clock rate	58.8MHz
Cycle/frame	46,000

5.3 Operation Frequency

The allocated time for processing a frame without the LSP process is given as

$$\frac{(cycle/frame) - (cycles\ of\ LSP\ process)}{23.2 - (processing\ time\ of\ LSP\ hardware/frame)}$$

Since the LSP process amounts 57% of the total computation, the required operation frequency of the embedded processor is given as

$$\frac{4,344,035 \times (1 - 0.57)}{23.2 - 46,000 \times \frac{1}{58,800}} = 83,328\ [\mathrm{Hz}]$$

Therefore, at the operation frequency of 84 MHz, the processor can decode the Ogg Vorbis streams in realtime. This operation frequency is much lower than 188MHz which is discussed in 3.1, and fulfills the required performance of ARM7TDMI.

6. Conclusion

This paper has described an architecture of an Ogg Vorbis decoder and its VLSI implementation, dedicatedly for portable audio appliances.

The LSP module has been implemented as a VLSI core through the use of Synopsys *Design Compiler*. This module with 9,740 gates operates at 58.8MHz. The required operation frequency of the processor using the LSP module is 84MHz, which can be achieved by ARM7TDMI.

Development is continuing on SoC (System-on-a-Chip) implementation of portable audio devices employing the proposed Ogg Vorbis decoder.

References

[1] ISO/IEC IS 11173-3, "Information Technology - Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbit/s - Part 3: Audio," (Aug. 1993).

[2] Ogg Vorbis RC2, *http://www.xiph.org/ogg/vorbis/index.html.*

[3] F. Itakura, "Line spectrum representation of linear predictive coefficients of speech signals," *Journal of Acoustic Society of America.*, vol. 57, pp. S35 (Apr. 1975).

[4] C. H. Wu and J. H. Chen, "A novel tow-level method for the computation of the LSP frequencies using a decimation-in-degree algorithm," *IEEE Trans. on Speech and Audio Processing*, vol. 5, no. 2, pp. 106–115 (Mar. 1997).

[5] W. Peng, W. Ser and M. Zhang, "Bark scale equalize design using warped filter," *IEEE Proc. on Acoustics, Speech, and Singal Processing*, vol. 5, pp. 3317–3320 (May 2001).

[6] T. Theide, etal.: "PEAQ — The ITU Standard for Objective Measurement of Perceived Audio Quality," *Journal of Audio Eng. Soc.*, vol. 48, no. 1/2 (Feb. 2000).