Real-Time Hardware Implementation of 3D Sound Synthesis

Sathwik GS
Dept. of ECE, NITK Surathkal, India
acesathwik@gmail.com

Barun Kumar Acharya
Dept. of ECE, NITK Surathkal, India
barunkumaracharya@gmail.com

Bilal Ali
Dept. of ECE, NITK Surathkal, India
bilalali261997@gmail.com

Deepu S. P.
Department of Electronics and Communication Engineering
National Institute of Technology Karnataka
Surathkal, Mangalore, India

Sumam David S.
Department of Electronics and Communication Engineering
National Institute of Technology Karnataka
Surathkal, Mangalore, India

***Abstract*—In this paper, hardware design and implementation to realize the effect of 3D sound with time-varying FIR filters are presented. 3D sound is a type of audio that encapsulates and recreates the effect identical to the way our ears normally experience. The spatial location of sound results in its three dimensional aspect. To synthesize it from a stereo recording, Head Related Transfer Functions (HRTFs), which describe the spectral behaviour of sounds coming from a particular direction are used. FIR filters derived from this transfer function are applied to the incoming sound, yielding spatial effect. The system was implemented using 180 nm technology libraries targeting an Application Specific Integrated Circuit (ASIC) and the functionality was validated in real-time on FPGA.**

***Index Terms*—Binaural recording, HRTFs, 3D audio, Real time hardware design**

I. Introduction

For several years binaural recording has been the most commonly used method to provide an immersive sound experience to improve and personalize listening to many types of audio, like music and movies. The method involves using artificial or dummy heads plugged in with microphones to record and reproduce the same over headphones. But this type of recording requires special equipment and is not as portable as any other two channel recording instrument. Head Related Transfer Functions (HRTF) describe how a sound from a specific point will arrive at the ear (generally at the outer end of the auditory canal). The brain has learned to recognize these signatures over time and when it hears a sound, it finds out the direction from its HRTF memory. A pair of HRTFs can be used to synthesize a binaural sound that seems to come from a particular point in space.

Several methods have been proposed over the years in different transfer domains to increase the computational speed of localizing a sound source. One of the most robust methods is the Generalised Cross Correlation (GCC) technique as discussed by Knapp and Carter [1], where along with basic cross-correlation, a weighting function is applied in the power spectrum. Among different weighting functions, the most popular one called Generalised Cross Correlation - PHAse Transform (GCC-PHAT) was introduced by Kwon et al. [2]. The authors used PHAT based weighting function in the frequency domain to analyse the incoming signals. Broeck and Bertrand [3] proposed a time-domain equivalent of the GCC-PHAT method and compared the complexity in both the domains. The authors observed that the time-domain method is computationally superior to the frequency domain for similar accuracy. The technology of surround sound systems was invented in the 1930s. The basic concepts of 3D sound systems are explained in detail by Wenzel [4]. Alongside, development in recording technology and signal processing techniques, 3D reproduction of sound was then realized based on HRTFs from source to the right and left ears as discussed by Chong-Jin and Gan [5]. The robustness analysis, localization characteristics, structural model for binaural sound synthesis and head related transfer functions are discussed by Brown and Duda [6]. All these methods help in the reproduction of 3D sound using HRTFs, but not in real-time. There was no focus on efficient computation to achieve the same. The literature on real-time hardware implementation and architectural details of HRTF based 3D sound synthesis systems are very limited. A general purpose Digital Signal Processor (DSP) based real-time implementation of a 3D sound synthesis system was presented by Kim et al. [7]. It was a complex system and may not be applicable for low power embedded applications. A FPGA based implementation of a similar approach was proposed by Fohl et al. [8]. The system was implemented using μBlaze softcore processor on Xilinx® Virtex-5 ML507 FPGA. HRTF filters with 512 coefficients each were used. The disturbance during filter switching was one of the major drawbacks associated with that design [8].

In this paper, an ASIC architecture is proposed to synthesize sounds using HRTFs. Methods to avoid the distortion during the filter transition are also presented. The functionality of the design was tested and validated in real-time by implementing it on an FPGA. The rest of the paper is organized as follows: Section II explains the algorithm behind the implementation. Section III describes the architectural and implementation details and Section IV analyzes the results obtained from tests and experiments conducted.

978-1-7281-9396-0/20/$31.00 ©2020 IEEE

II. Algorithm

Since the convolution of the audio signals with HRTF corresponding to a particular direction makes it seem as if the sound source is placed in that spatial area, it is important to locate the source. This is done by cross-correlating the left and right channels. The delay estimated by cross-correlation is mapped to an angle that approximates the angular position of the source. From the derived angle, the FIR filter corresponding to the same is chosen.

As illustrated in Fig. 1, the received audio signals on left channel *l(t)* and right channel *r(t)* are cross-correlated to estimate the delay or the time difference of arrival of signals. The angle of the sound source is then estimated as follows:

$$\tau = \frac{\text{Max}(l(t) * r(t)) - \text{Signal length}}{f_s} \quad (1)$$

where τ is the delay between the two signals in seconds, 'Signal length' is the correlation frame length, f_s is the sampling rate and '*' is the cross-correlation operation whose index of the maximum value is returned.

$$\sin\theta = \frac{\tau c}{d} \quad (2)$$

where c is the speed of sound, i.e., 343 m/s, d is the distance between two microphones and θ is the azimuthal angle estimated. Only the index of maximum cross-correlated value is determined. The angle θ is calculated using a Look-Up Table (LUT), generated with each index mapped to a pre-calculated angle. Given that the HRTFs database [9] has 25 different azimuths, by using the θ derived, a filter corresponding to the closest angle is chosen.

III. RTL Design

The top level block diagram of the system is shown in Fig. 1. The designed system records samples using microphones separated by 20 cm (approximate distance between the ears) at a sampling rate of 44.1 kHz. Filters from one of the most popular databases - CIPIC HRTF database [9] are used to convolve any incoming audio signal. The original filter length of 200 coefficients was down sampled in the frequency domain and converted to a 100 tap filter to reduce the number of computations, but not reduced further to preserve the frequency resolution. Data was represented using the signed 16-bit fixed-point arithmetic.

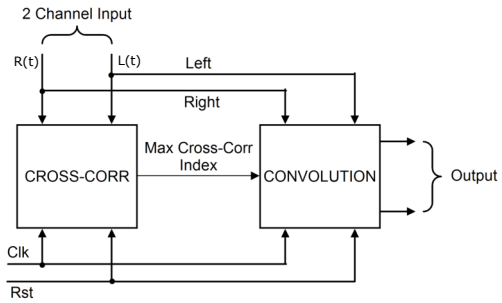

Fig. 1. Top Level Block Diagram

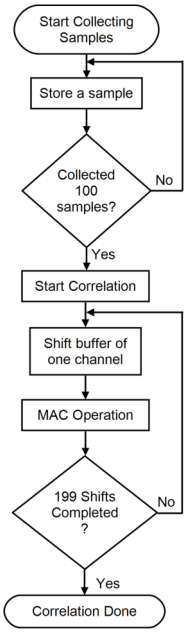

Fig. 2. Flowchart of the Correlation Computation

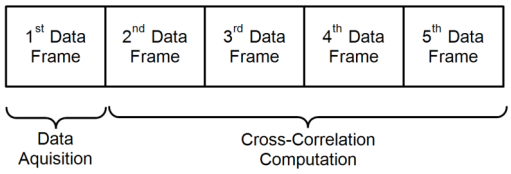

Fig. 3. Pipeline for cross-correlation

A. Cross-Correlation Module

This unit cross-correlates data buffers 100 samples long and returns the index of maximum value. One of the buffers is extended to a length of 298 samples but non-zero samples are present only from index 100 to 199, rest are filled with zeros. This is done so that only the smaller buffer is slid over the other corresponding to each shift. The correlation is done by performing a MAC operation on each shift as illustrated in Fig. 2 and the whole operation is divided into two parts: data acquisition and correlation computation. The samples are collected at every 5^{th} data frame and the computation is done over a period of 4 data frames as shown in Fig 3. Each shift in the computation requires 100 MAC operations, hence for 199 shifts we require 19900 operations to be completed in the time period of 4 data frames, i.e., 400 sample periods. The clock speed required for a MAC operation is at least 50 times faster than the sampling rate. This corresponds to having 50 MAC operations per sample period with 100 clock periods as idle time in the end.

B. MAC Unit

This is the only arithmetic unit of the design. It uses one multiplier and adder, as shown in Fig 4. The 32-bit result is truncated and stored in a 16-bit register.

C. Convolution Module

Every data frame of 100 samples is convolved with a filter chosen according to the index given by the correlation module as shown in Fig. 5. Each output sample is computed by 100 MAC operations. Even though the correlation module requires only 50 times faster clock speed, this module requires a 100 times faster clock, i.e., 4.41 MHz. Not every result of the cross-correlated index is mapped to a filter and convolved, instead, past 10 indexes are stored and the filter which is mapped the most number of times is finally chosen, as illustrated in Fig 6. This means that there is a possibility of a filter change at every 50^{th} data frame, i.e., at about 110 ms which can account for any change in the position of the sound source.

D. Convolution during filter change

For 50 frames of data, a fixed filter is used after which there can be a change in the filter. Changing the entire filter buffer with the new set of coefficients would be incorrect since convolution with the current filter would be incomplete. Two convolution units are required at the same time, one - to finish the ongoing convolution process and the second - to begin filtering with the new coefficients.

However, this requirement can be avoided by carefully observing the impact of coefficients on the output. We use the *stepwise replacement of impulse response* method as mentioned in [10] but applied in time domain to give the same result. The coefficients are replaced in the filter buffer at the same input sampling rate.

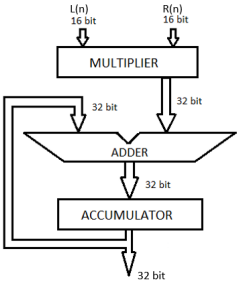

Fig. 4. MAC Unit

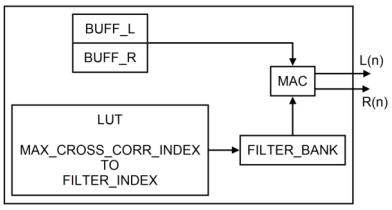

Fig. 5. Convolution Module

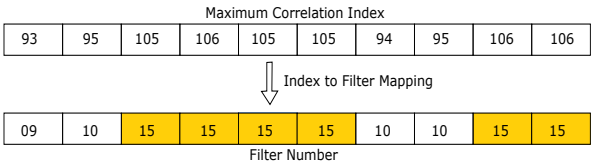

Fig. 6. Example of Filter Selection

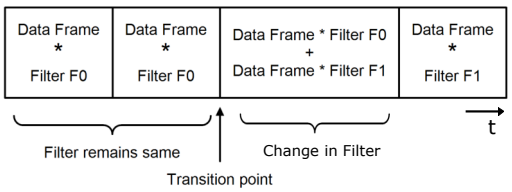

Fig. 7. Transition between filters

E. Smooth transition

To avoid audible glitches during the transition, careful switching of filters is done , followed by complete replacement of current filter. The sequence of transition is illustrated in Fig. 7, showing the impact of filters on the output .
For $0 < n < n_l$

$$\text{h\_buffer}(n) = h\_new(n) \tag{3}$$

for $n_l < n < 100$

$$\text{h\_buffer}(n) = h\_curr(n) \tag{4}$$

where the value of $0 < n_l < 100$ increments at sampling rate, indicating the number of coefficients replaced so far, *h_new* and *h_curr* contains the filter coefficients of new filter chosen and current filter respectively with *h_buffer* holding the overall impulse response.

This implementation makes sure that only one convolution unit is used all the time along with the handling of filter transition without any abrupt change in the output.

IV. Implementation Results

The synthesis of the design was done with 0.18 μm Semi-Conductor Laboratory (SCL) standard cell libraries using Cadence® Design Suite. To verify the functionality of detecting the angle of the sound source, recorded samples were given as input in the post synthesis simulation of the design. Samples were recorded with National Instruments Data Acquisition device at a sampling rate of 44.1 kHz. The experiment involved placing the sound source at 40° to the right from the center of the microphones. Fig. 8 indicates that the source was placed at an angle corresponding to index 116. Using equation (1) and (2) this index maps to angle = 38.47° Although the module resulted in other indexes but having a scheme as shown in Fig. 6 helps in minimal acceptance of those. Another experiment involved moving the source continuously from left to center (and vice versa). Fig 9 shows the resulting indexes. The resultant mapping ranges from 30° to the left to center. Apart from the central position, most of the angles occur equally likely, indicating constant movement.

To test the design and check its functionality in real-time, Xilinx Nexys 4 DDR Artix -7 FPGA was used. Initially, experiments were conducted to realise the effect of HRTFs by

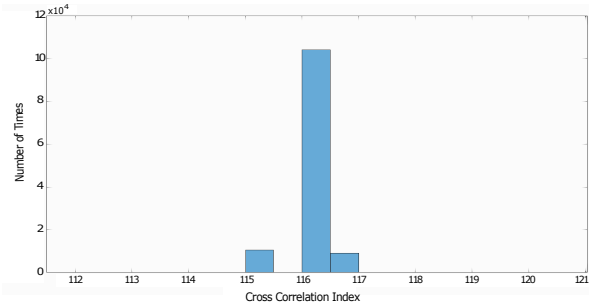

Fig. 8. True angle of source = 40°

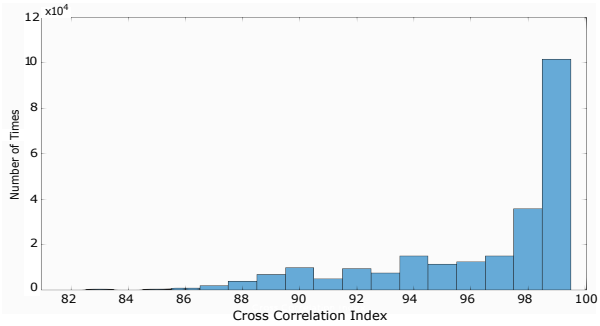

Fig. 9. Continuous movement of source ranging from left to center

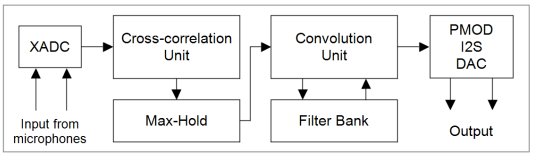

Fig. 10. Block Representation

manually changing the filters to replicate the panning effect. ADMP401 MEMS microphones which are omnidirectional, low power and have a flat frequency response from 100Hz to 15kHz were used as input (analog). The *XADC* present in Nexys 4 was used to convert the microphone output to a 16-bit digital signal. The output is converted back to its analog equivalent using *PMOD I2S DAC* module which uses the I2S protocol.

After checking the functionality by manually changing the filters, the proposed design which automatically detects the angle of source and chooses the appropriate filter as shown in Fig 10 was implemented. The resource utilization is given in Table I. Here, all the filter coefficients were directly stored in registers, taking up most of the resources. As expected, the RTL design implemented was able to locate the sound source. The movement of source was accurately tracked and helped in filter selection. Subjecting to the listening test, users were able to easily distinguish different directions.

V. Conclusion

A real-time hardware implementation of 3D audio using binaural processing was done and its various characteristics were discussed in this paper. Head Related Transfer Functions from the CIPIC HRTF database was used for realising the 3D effect. Functional validation was done through post synthesis simulation and real-time testing conducted on FPGA. The design and algorithm worked as expected enabling the user to realise 3D sound via headphones. The algorithm can be further optimised according to different applications depending on specific requirements by exploring other available HRTF databases.

TABLE I
Resource Utilization

Resource	Utilization	Available	Utilization %
LUT	7957	63400	12.55
FF	1678	126800	1.32
DSP	2	240	0.83
IO	36	210	17.14
BUFG	8	32	25.00
MMCM	1	6	16.67

References

[1] C. Knapp and G. Carter, "The generalized correlation method for estimation of time delay," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 4, pp. 320–327, August 1976.

[2] B. Kwon, Y. Park, and Y. Park, "Analysis of the gcc-phat technique for multiple sources," in *ICCAS 2010*, Oct 2010, pp. 2070–2073.

[3] B. Van Den Broeck, A. Bertrand, P. Karsmakers, B. Vanrumste, M. Moonen *et al.*, "Time-domain generalized cross correlation phase transform sound source localization for small microphone arrays," in *2012 5th European DSP Education and Research Conference (EDERC)*. IEEE, 2012, pp. 76–80.

[4] E. M. Wenzel, "Localization in virtual acoustic displays," *Presence: Teleoperators & Virtual Environments*, vol. 1, no. 1, pp. 80–107, 1992.

[5] C.-J. Tan and W.-S. Gan, "User-defined spectral manipulation of hrtf for improved localisation in 3d sound systems," *Electronics Letters*, vol. 34, no. 25, pp. 2387–2389, Dec 1998.

[6] C. P. Brown and R. O. Duda, "A structural model for binaural sound synthesis," *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 5, pp. 476–488, Sep. 1998.

[7] H.-J. Kim, D.-G. Jee, M.-H. Park, B.-S. Yoon, and S.-I. Choi, "The real-time implementation of 3d sound system using dsp," in *IEEE 60th Vehicular Technology Conference, 2004. VTC2004-Fall. 2004*, vol. 7. IEEE, 2004, pp. 4798–4800.

[8] W. Fohl, J. Reichardt, and J. Kuhr, "A system-on-chip platform for hrtf-based realtime spatial audio rendering with an improved realtime filter interpolation," *International Journal on Advances in Intelligent Systems*, vol. 4, no. 3, pp. 309–317, 2011.

[9] V. R. Algazi, R. O. Duda, D. M. Thompson, and C. Avendano, "The cipic hrtf database," in *Proceedings of the 2001 IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics (Cat. No.01TH8575)*, Oct 2001, pp. 99–102.

[10] Ø. Brandtsegg and S. Saue, "Live convolution with time-variant impulse response," *Proceedings of the 20th International Conference on Digital Audio Effects*, pp. 239–246, Sep. 2017.