Independent Component Analysis applied to watermark extraction and its implemented model on FPGAs

T. Le-Tien, D. Vo-Ngoc, L. Ngo-Hoang, SungYoung Lee*
Dept. of Electrical Electronics Engineering, *Dept. of Computer Engineering
Ho Chi Minh City University of Technology, Vietnam; *KyungHee University, Korea
thuongle@hcmut.edu.vn, hoanglane@yahoo.com, vongoc_pfiev@yahoo.com, sylee@oslab.khu.ac.kr

***Abstract**:* Most of published audio watermark algorithms are suffered a trade-off between inaudibility and detectibility, and the detection performance depends greatly on the strength of noise added by communication channels. This work introduces an audio watermarking method that can overcome this challenge, i.e. allows increasing watermark strength while preserving inaudibility. The scheme uses psychoacoustic masking compatible to MPEG layer 1 Model 1 and adjusts it in a data adaptive way. A blind watermark extraction technique using the Independent Component Analysis (ICA) is shown to minimize the watermark decoding error. An implementation of a simple quantization-based watermarking algorithm (LSB) on the Spartan-3 FPGA Starter Kit of Xilinx is also presented as a part of hardware demonstration of the method.

Keywords: Audio watermarking, LSB, Psychoacoustic masking, ICA, FPGA, Spartan-3.

I. INTRODUCTION

Watermarking is a method to embed sideband data such as author copyright information into the original signal. There are two significant groups named video and audio watermarking which has been now studied and developed in world-wide.

Many techniques have been proposed for audio watermarking, in both time and frequency domains [1,3]. Although most of these techniques behave well in theoretical domain, their performance falls down strongly over communication channels. Our works research a frequency-domain audio watermarking technique which can minimize the decoding error by controlling the watermark accuracy at the encoder stage. The method is compatible to MPEG Layer 1 Model 1 and robust to stereo-to-mono conversion and
Gaussian noise as well. It applies Psychoacoustic Masking in a data adaptive way, thus allows increasing watermark strength while preserving inaudibility of the watermark, which has been often a trade-off for the others. In addition, this work proposes an audio watermark extraction technique which adopts Independent Component Analysis (ICA), FastICA algorithm particularly, for blind watermark decoding. It does not require the original data for decoding, except the secret key. The method is not only a watermark extraction tool, but also a synchronization tool between the transmitter and the receiver. Any embedding process can be used, in which the original audio and watermark are statistically independent and linearly mixed. All these algorithms are implemented in Matlab. Furthermore, we carry out some demonstrations of LSB algorithm, a quantization-based watermarking method, on Spartan-3 Starter Kit of Xilinx.

II. ADAPTIVE AUDIO WATERMARKING USING PSYCHOACOUSTIC MODEL 1

Our watermarking scheme uses the masking model designed in ISO/IEC 11172-3 (MPEG-1, layer 1) Audio Psychoacoustic Model. Audio Masking is the effect by which a faint but audible sound becomes inaudible in the presence of another louder audible sound, i.e., the masker. The masking effect depends on the spectral and temporal characteristics of both masked signal and the masker. Frequency masking refers to masking between frequency components in the audio signal, frequency components remaining below the masking threshold are suitable places for watermark embedding. Frequency masking effect can be illustrated in Figure 1 [3,10].

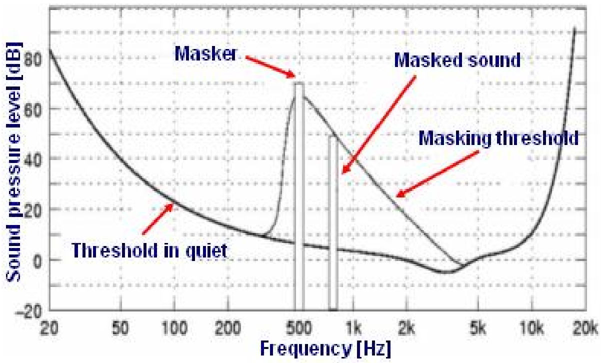


Figure 1. Frequency masking

Step 1: Calculate the Spectrum

Each frame of audio signal with length of N=512 is weighted with a Hanning window. The power spectrum is calculated as:

$$S(k) = PN + 10*\log_{10}\left|\sum_{n=0}^{N-1} s(n)h(n)\exp\left(-j2\pi\frac{nk}{N}\right)\right|^2 \quad (1)$$

where $0 \le k \le \frac{N}{2}, PN = 90.302dB$ and the Hanning window, $h(n)$, is defined as

$$h(n) = \frac{1}{2}\left[1-\cos\left(\frac{2\pi n}{N}\right)\right] \quad (2)$$

Step 2: Identify Tonal Components

A frequency f (with FFT index k) is a tone if its power S(k) is:

- Greater than S(k-1) and S(k+1)
- $S(k) - S(k+j) \geq 7dB$

$$\begin{cases} j \in [-2,+2] \text{ if } 2<k<63 \\ j \in [-3,-2,+2,+3] \text{ if } 63 \leq k<127 \\ j \in [-6,...,-2,+2,...,+6] \text{ if } 127 \leq k \leq 250 \end{cases} \quad (3)$$

Step 3: Remove Masked Components

Components below the absolute hearing threshold and tonal components separated by less than 0.5Barks are removed. The absolute hearing threshold is well approximated by nonlinear function:

$$S_a(f) = 3.64(f/1000)^{-0.8} - 6.5e^{-0.6(f/1000-3.3)^2} + 10^{-3}(f/1000)^4 \quad (\text{dB SPL}) \quad (4)$$

A frequency in Hertz can be converted into the bark scale using the function:

$$z(f) = 13\arctan(.00076f) + 3.5\arctan\left(\frac{f}{7500}\right)^2 (\text{Bark}) \quad (5)$$

Step 4: Calculation of individual tone and noise masking thresholds $P_t(f_i, f_t)$ and $P_n(f_i, f_n)$ which can be derived in [10]

Step 5: Calculation of global Masking thresholds

The global masking threshold for each frequency f_i takes into account the absolute hearing threshold S_a and the masking curves P_i of the N_t tonal components and N_n non-tonal components:

$$S_m(f_i) = 10 * \log_{10}\left[10^{0.1S_a(f_i)} + \sum_{t=1}^{N_t} 10^{0.1P_t(f_i,f_t)} + \sum_{n=1}^{N_n} 10^{0.1P_n(f_i,f_n)}\right] \quad (6)$$

where f_j denotes the set of frequencies in the signal.

As a result, for each audio block of N=512 samples, a masking value for each frequency component is produced. Our watermarking scheme uses Psychoacoustic Model and allows minimizing the decoding error by using a feedback loop at the encoding stage. With this feedback loop, a watermark bit is inserted repeatedly until there is no error between that bit and the estimated watermark bit at the encoder. In this paper, audio data is sampled at 44100 Hz, 16bits and each frame has the length of N=512 samples. The watermark takes the form of text which can be encoded using ASCII code, thus sequence of bits 1 and 0 is generated. Bits 1 are retained while bits 0 are substituted by bits -1. Therefore, the watermark bit can be 1 or -1. The secret key is used to generate a PN sequence with zero-mean. So k is a 512-bit vector of values 1 or -1 (substitute for 0).

The audio data is processed frame by frame. Let s_i^t refers to the i^{th} audio frame at state *t*. This adaptive watermarking method requires iterative insertion of one watermark bit, w_i, into each audio frame, thus let *t* be the counter of these insertions thus refers to the state of insertions. The initial value of *t* is equal to one. Thus each watermark bit is embedded into the audio data frame at least once. The encoder embeds one watermark bit w_i into each frame, i can be any integer from 1 to the length of watermark. At each instant, the encoder takes an original audio frame, s_i^t as its input and transmits the corresponding watermarked frame, $s_{i_{WM}}^t$, over the communication channel. Watermark embedding is performed by:

$$s_{i_{WM}}^t = s_i^t + f(s_i^t, w_i)k = s_i^t + w_i k_{m_i} \quad (7)$$

$f(s_i^t, w_i)$ is a nonlinear function of s_i^t and the watermark bit w_j. k_{m_i} is modulated key generated in frequency domain by applying Psychoacoustic Masking to k.

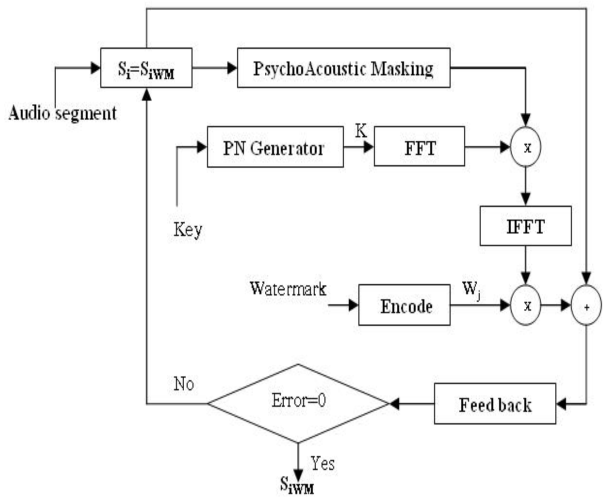


Figure 2. Block diagram of watermarking encoder

Figure 2 presents the block diagram of the proposed watermarking encoder. The block Psychoacoustic Masking produces masking threshold in a data adaptive way. The watermark embedding technique embeds a modulated secret key, k_m, into each frame while inserting one watermark bit per frame as stated previously. Data embedding is performed in the time domain and the sign of the modulated key is specified by the inserted watermark bits. The proposed WM encoding scheme allows using maximum embedding capacity in a data adaptive way to make sure that the decoding accuracy is high. Our scheme minimizes decoding error at encoding stage by using a feedback loop. At each frame out, S_{iWM}, a feedback checks the WM decoding error. If the error is not equal to zero, it means that correct detection of watermark bit is impossible, then the watermarked audio frame is re-watermarked. This idea is to increase WSR (Watermark to Signal ratio) by iterative insertions. Because at each state, the embedding capacity will vary depending on the masked original audio. Each re-embedding allows us to change masking thresholds in a data adaptive way, thus to keep the inserted watermark just below the audible level while increasing the insertion capacity. Decoding error $\hat{e}$ is described as the relationship between the estimated watermark bit $\hat{w}_i$ and the original water-mark bit w_j.

$$\hat{e} = \hat{w}_i - w_i \quad (8)$$

With: $\hat{w}_i$ = sign(r) and r is the correlation between the modulated key and the watermarked audio signal at the according state.

$$r = \frac{\sum_{n=0}^{N-1}\left(k(n)-\langle k\rangle\right)\left(s_{i_R}^t(n)-\langle s_{i_R}^t\rangle\right)}{\sqrt{\sum_{n=0}^{N-1}\left(k(n)-\langle k\rangle\right)^2}\sqrt{\sum_{n=0}^{N-1}\left(s_{i_R}^t(n)-\langle s_{i_R}^t\rangle\right)^2}} \quad (9)$$

where $\langle . \rangle$ is the mean operation,

N is the sample number (here N = 512).

The advantage of this watermarking scheme is that it can increase the watermark strength while preserving the quality of original audio signal. Furthermore, in this embedding process, the embedded key sequence and the original signal can be considered as two independent sources which are linearly mixed, as can be seen in Equation 7, and then we can use ICA (Independent Component Analysis) to extract the watermark. This subject will be discussed in the next section.

III. ICA WATERMARK EXTRACTION

Independent Component Analysis (ICA) attempts to separate a set of observed signals that are composed of linear mixtures of a number of independent non-Gaussian sources into a set of signals that contain the independent sources [2]. Early work in this area was carried out by Comon [5], and Bell and Sejnowski [4]. Recent years have seen an explosion in interest in ICA, with an ever increasing number of algorithms being presented that perform independent component analysis, such as the FastICA algorithm [2], Jade ICA [11] and the kernel approach taken by Bach [6].

The underlying mathematical model for ICA can be stated as follows. Assume that there are N independent sources, s_i, which transmits signals which are measured by M sensors. The signals measured by the sensors, x_i, can be mapped to the sources using an unknown function f_i, resulting in:

$$x_i = f_i\left(s_1, \ldots, s_N\right) \quad (10)$$

It is assumed that the contributions of each of the N sources add together linearly to create each x_i. Using matrix notation, the equation can be written in a more elegant form:

$$x = As \quad (11)$$

with: $x^T = \left[x_1, \ldots, x_M\right]$, $s^T = \left[s_1, \ldots, s_N\right]$

and **A** is an M x N invertible matrix, called the mixing matrix. In most ICA algorithms, the number of sensors has to equal the number of sources, resulting in **A** being of size N x N. ICA then attempts to estimate the matrix **A**, or, equivalently, to find an unmixing matrix **W** such that:

$$y = Wx = WAs \quad (12)$$

gives an estimate of the original source signals where $y^T = \left[y_1, \ldots, y_N\right]$, and **W** is of size N x N.

The matrix **y** will have independent components y_i if and only if:

$$p(y) = \prod_{i=1}^{N} p(y_i) \quad (13)$$

where $p(y_i)$ is the probability density function (pdf) of y_i, and $p(\boldsymbol{y})$ is the joint pdf of the matrix $\boldsymbol{y}$.

ICA seeks to find an unmixing matrix **W** such that the resulting matrix **y** has component pdfs that are factorisable in the manner shown in equation (13). It is possible to obtain such an unmixing matrix given two constraints, these being that we cannot recover the source signals in the order in which they came in, and we cannot get the original signals in their original amplitude [2]. The key to solve ICA problem is the nongaussianity of the original sources. ICA attempts to estimate w which is one of the rows of the matrix W in such a way that maximizes the nongaussianity of $w^T x$. The FastICA algorithm is a fixed-point algorithm for carrying out ICA. It is based on the use of negentropy as a cost function. Negentropy is defined in Equation 13, always non-negative and is zero only for a Gaussian variable.

$$J(y) = H(y_{gauss}) - H(y) \quad (14)$$

where y_{gauss} is a Gaussian random variable of the same covariance matrix as y.

Approximations to negentropy are normally used to simplify the estimation. The family of approximations to negentropy used in FastICA is:

$$J(y) \approx \sum_{i=1}^{p} k_i \left[E\left\{G_i(y)\right\} - E\left\{G_i(v)\right\}\right]^2 \quad (15)$$

where k_i are some positive constants, v is a Gaussian variable of zero mean and unit variance, and E is the expectation of the function. The random variable y_i is assumed to have zero mean and unit variance. G can be nearly any nonquadratic function which is chosen with regards to the robustness of the estimates of negentropy obtained from the function. The functions chosen in [2] were:

$$G_1(y) = \frac{1}{a_1}\log\cosh a_1 y, G_2(y) = -\mathrm{e}^{-a_2 y^2/2} \quad (16)$$

where a_1 and a_2 are constants in the range of 1 to 2. Using these functions, an iterative algorithm was designed by Hyvaerinen [2] for implementing ICA. The algorithm is as follows:

1. Choose an initial weight vector w.
2. Let $w^+ = E\left\{xg(w^T x)\right\} - E\left\{g'(w^T x)\right\} w$
   where g is the derivative of function G and g' is the derivative of g.

3. Let $w = w^{+} / \left\| w^{+} \right\|$
4. Decorrelate the output to prevent the vectors from converging to the same maxima by letting $w = \left(ww^{T} \right)^{-1/2} w$.
5. If not converged, go to step 2.

FastICA, as its name suggests, runs faster than other ICA algorithms when dealing with large batches of data without any compromises in the performance of the algorithm. In our work, the watermarking scheme presented in section 2 is used for watermark embedding. The scheme can be described as a process of embedding a secret key sequence, specified by the original audio and the watermark bit stream, into the original audio stream. Hence, we can consider the key sequence and the original signal as two independent non-Gaussian sources. ICA is used to separate the secret key from original audio data by making process on their received observations. Then, by calculating the correlation between the decomposed key and the secret key, the watermark bit w decoded as 1 or -1. Since ICA implementation needs a number of mixtures at least equal to that of original sources, it is required that two observations of watermarked noisy data sequences are recorded at the receiver. The proposed watermark extraction technique first applies the whitening process on the received inputs by using PCA (Principle Component Analysis). Then ICA block performs decomposition on the whitened audio resulting in the estimated embedded key and the original signal. This is followed by a correlation based traditional decoding scheme. Note that here the audio data is processed frame by frame. Figure 3 shows the block diagram of this watermark decoder [9].

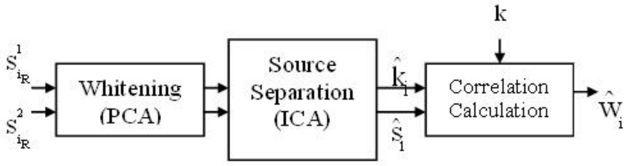


Figure 3. Watermark decoder using ICA

where $S^{1}_{i_R}$, $S^{2}_{i_R}$ refer to received watermarked signals of ith audio frame. $\hat{S}_i, \hat{k}_i$ are the estimation of the original signal s and the secret key k embedded in ith frame. $\hat{w}_i$ denotes the estimated watermark bit.

Let $S^{t}_{i_R}$ refers to the received watermarked signal as tth observation, $t = 1,2,...,m$ where $m \geq 2$. The PCA based whitening process is applied on $S^{t}_{i_R}$ to guarantee the independence of observations. It gives uncorrelated, unit variance signals $\tilde{S}^{t}_{i_R}$, $t = 1,2$ as below,

$$\tilde{S}^{t}_{i_R} = ED^{-1/2} E^{T} S^{t}_{i_R} \tag{17}$$

where E is an m by m (here $m = 2$) orthogonal matrix of eigenvectors of covariance matrix $E\left\{ S^{t}_{i_R} S^{t\,T}_{i_R} \right\}$ and D is an m by m diagonal matrix of its eigenvalues.

In this application, PCA not only estimates uncorrelated components but also help us to detect whether the received audio streams are watermarked or not. It is due to the variance based characteristic of PCA; thus if one of or both of these audio streams are not watermarked, the dimension of data is automatically reduced to one.

ICA then takes the whitened $S_R = \left[\tilde{S}^{1}_{i_R}, \tilde{S}^{2}_{i_R} \right]$ as its inputs and separates them into independent source signals $S_S = \left[S^{1}_{i}, S^{2}_{i} \right]$ by estimating the inverse of unknown full-rank mixing matrix $W = A^{-1}$ as presented previously. Here we use FastICA algorithm to carry out the estimation of W. After that, multiplying W and S_R together will result in S_S.

$$S_S = WS_R \tag{18}$$

According to our model, one of the two S_S elements is the estimated secret key $\hat{k}_i$ and the other is the transmitted original audio signal. Note that $\hat{k}_i$ is not a noise free signal due to the effect of the additive channel noise on it. Since the watermark extraction scheme is blind, the only signal available at the decoder is the secret key k. We use this k to detect the transmitted watermark bit by carrying out the correlation between k and decomposed source signals as follows:

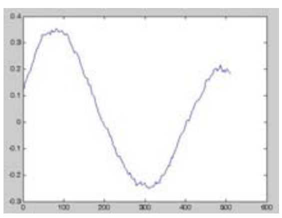
$$r(\tau) = \sum_{l=1}^{M} k(l) S_i^{t}(l+\tau) \tag{19}$$

where $t = 1, 2,\ l = 1, .., 512,\ \tau = 0,1,...,L$.

Ideally, the correlation should be equal to zero for the original audio stream. In the other hand, it becomes maximized in absolute value for the estimated embedded key. By choosing this maximum correlation r, the inserted watermark bit can be extracted according to the sign of r:

$$\hat{w}_i = \begin{cases} 1 & sign(r) > 0 \\ -1 & sign(r) < 0 \end{cases} \tag{20}$$

IV. SIMULATION RESULTS



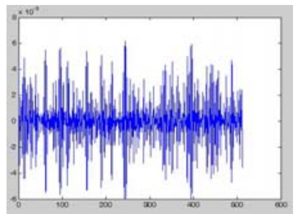
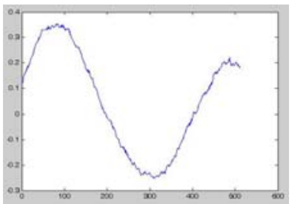


Figure 4. Original frame (Left)
Figure 5. Key modulated K_m (Right)

Test data is a wave file sampled at 44100 Hz, 512 samples per frame, thus the length of an audio frame is 11msec. These results obtained by applying psychoacoustic masking and watermarking to the first audio frame. We notice that in our encoder scheme, the masking threshold is used as a filter to color the pseudo-random noise sequence. The colored sequence is then scaled to be below the threshold. The outcome is an inaudible noise sequence. So we can consider K_m as a noise sequence and it becomes inaudible when embedding into original frame. In encoder we also use a feed back to reduce error at the decoding, so each frame can be re-watermarked many times. It is shown in Figure 7 that, watermark insertion repeat number varies from frame to frame.



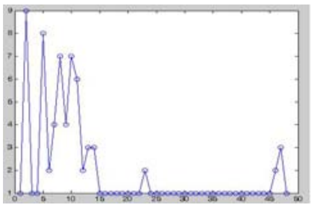
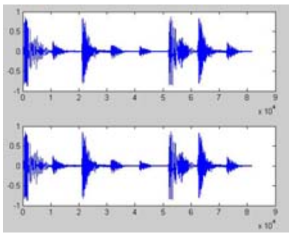


Figure 6. Watermarked Frame (Left) and,
Figure 7. Distribution of watermark insertion repeats versus frame number for 48 frames.

In order to guarantee the extraction of water-mark signal by ICA, it is required that the number of observed inputs is at least equal or larger than the number of independent audio sources. Thus we need to acquire at least two watermarked noisy data sequence at the receiver site. Here we can generate two ICA inputs by alternating the watermark strength. It is performed in Figure 8 the case in which one is watermarked with normal strength of the watermark and the other with doubled.



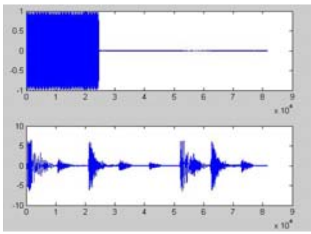


Figure 8. Wave forms of two inputs of ICA implementation (Left)
Figure 9. Wave forms of embeded key and the original signal (Right)

Separation result of ICA implementation in Fig.9 shows that the two decomposed signals estimate the embedded key and the original signal, respectively. The estimated key has a length larger than that of the original secret key; this is easy to understand because ICA give two signals with equal lengths, but the important thing is that the non-zero part of the estimated key is likely to the embedded key in both shape and length. An encountered problem when using ICA to extract the watermark for audio signals is that in some cases, the whole extracted bits are inverted (-1 to 1 and vice versa). This is due to the ambiguity of the sign of ICA. To overcome this problem, we propose two solutions. One way is that we use one synchronization bit to determine the sign of the watermark bit stream. The decoding stream finds out the exact value based on the sign of this bit. Another way is that we use two different secret keys at the encoding stage, one key is for bits 1 and the other is for bits –1. At the decoding stage, the both secret keys are used for the correlation calculation. For each frame, if the embedded bit is 1, then only the correlation operation for the according secret key is converged, and the same for bits –1. Hence, we can estimate the whole watermark bits exactly, with a lower bit rate.

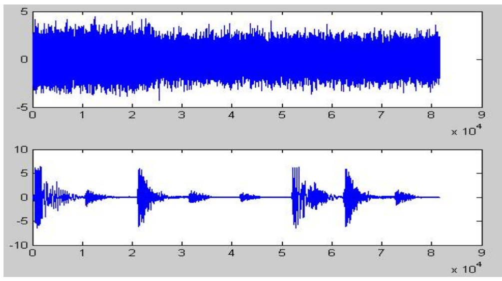


Figure 10. Decomposed waveforms, SNR of 35dB

We also process the simulation with additive channel noise for this model. The result demonstrates that the watermark has a high robustness against Gaussian noise, a desirable thing for almost all published algorithms. The watermark extraction is still obtained exactly with a signal-to-Gaussian noise ratio about 32-35dB for this method, while about 50-60dB for the others. Fig.10 shows the result of ICA decomposition when watermarked data is attacked by 35dB-Gaussian Noise. As can be seen in Figure 10, the estimated key is terribly deformed when 35-dB Gaussian noise is added, but this fact does not affect the result of watermark estimation.

V. LSB ALG. IMPLEMENTED ON FPGA

The Xilinx Spartan-3 Starter Kit provides a low-cost, easy-to-use development and evaluation platform for Spartan-3 FPGA designs [8]. Taking the advantage of this Spartan-3 development system on hardware simulation, our work attempts to carry out an implementation of LSB (Lowest Significant Bit) algorithm, a quantization based audio watermarking method, on the card.

The fundamental principle of LSB method is very simple. It replaces the lowest bit in each audio frame by the watermark bit. The value of these watermark bits can be 0 or 1. Our work proposes a model of LSB algorithm implemented on Spartan-3 FPGA card by following steps:

Step 1: The audio data are sampled by PC and transmitted to FPGA card through COM port.

Step 2: FPGA card receives and saves these data under the format of 32-bit binary numbers.

Step 3: The processing block reads 32-bit data samples from RAM as well as the watermark bits set by the switch on FPGA card.

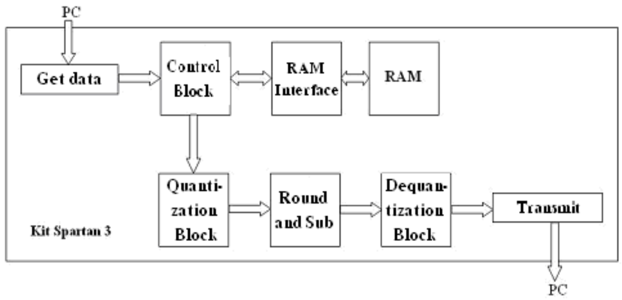


Figure 11. LSB implementation on FPGA

Step 4: For each sample, find the bit of each 32-bit data sample according to the 16th bit of the same data's 16-bit sample. Then replace that bit by the appropriate watermark bit. Go to next samples and repeat the replacement until the end of the watermark bit stream.

Step 5: Dequantize those watermarked samples and return the results to PC also through COM port.

Step 6: The program on PC receives data and records them into a watermarked audio file.

Figure 12. illustrates the block diagram of LSB implementation on Kit FPGA. The computation on Kit is done on floating point IEEE 754. The value of an IEEE-754 number is computed as:

$$N = (-1)^{s} * 2^{E-127} * 1.M \qquad (21)$$

TABLE 1. 32-bit IEEE-754 floating point format

Fction M (23 bits)	Exponent E (8 bits)	Sign s (1 bit)

As shown in Table 1, the sign is stored in bit 32. The exponent can be computed from bits 24-31 by subtracting 127. The mantissa (also known as fraction) is stored in bits 1-23. An invisible leading bit (i.e. it is not actually stored) with value 1.0 is placed in front, then bit 23 has a value of 1/2, bit 22 has value 1/4 etc.

As a result, the mantissa has a value between 1.0 and 2. If the exponent reaches -127, the leading 1 is no longer used to enable gradual underflow. In our work, we use Xilinx 7.1i software and VHDL language [7] to construct the schematic as shown in Figure 13. Watermark Sequence is generated by switches on Kit. Pushing on switch gives bit ‘1’ and pulling off switch gives bit ‘0’. The result will be saved to PC and the watermarked signal is written into a wave file. We have created a graphical user interface using Matlab to control transmission and receive data between PC and FPGA Kit.

VI. CONCLUSION

In this paper, we propose a data adaptive watermark encoder using Psychoacoustic Model that allows minimizing the decoding error at the encoder. An application of the ICA method to extract watermark for audio signals is also presented. The simulation results show that by using Fast ICA algorithm we can extract watermark exactly. In our work, we have succeeded in precisely extracting the watermark for embedding process using Psychoacoustic Model under Gaussian Noise attack of 35dB.

Furthermore, we also study the Spartan-3-FPGA card from Xilinx and implement LSB watermarking algorithm on the Kit as a part of the analyzed model. A future developed work will be processed for whole system on the FPGAs and multiple component inputs are tested with the method to obtain a general approach for ICA implementation.

References

[1] Arnold, M., Schmucker, M. & Wolthusen, S., *Techniques and Applications of Digital Watermarking and Content Protection*, Artech House, Boston, **2003**.

[2] Hyvaerinen, A. & Oja, E., Independent Component Analysis: Algorithms and Applications, *Neural Networks*, 13(4-5), **2000**, pp. 411-430.

[3] Swanson, M.D., Zhu, B., Tewfik, A.H. & Boney, L., Robust Audio Watermarking Using Perceptual Masking, *Signal Processing*, vol. 66, **1998**, pp. 337-355.

[4] Bell, A.J. & Sejnowski, T.J., An information-maximisation approach to blind separation and blind deconvolution, *Neural Computation*, 7, **1995**, pp. 1129-1159.

[5] Comon, P., Independent component analysis - a new concept, Trans. On *Signal Processing*, 36, **1994**, pp. 287-314.

[6] Bach, F.R. & Jordan, M.I., Kernel Independent Component Analysis, *Journal of Machine Learning Research*, 3, **2002**

[7] Perry, D.L, *VHDL Programming by Example*, 4th Ed., McGraw-Hill, New York, **2002**.

[8] Spartan-3 Starter Kit Board User Guide, available at www.Xilinx.com

[9] Sener, S. & Gunsel. B., Blind Audio watermark Decoding Using Independent Component Analysis, *Proc. of the 17th International Conf. on Pattern Recognition (ICPR’04)*, **2004**.

[10] Painter, T., & Spanias, A., Perceptual Coding of Digital Audio, *Proceeding of the IEEE*, Vol. 88, April **2000**, pp. 451-515.

[11] Cadoso, J.F., Jade ICA, available at www.gerstner.felk.cvut.cz

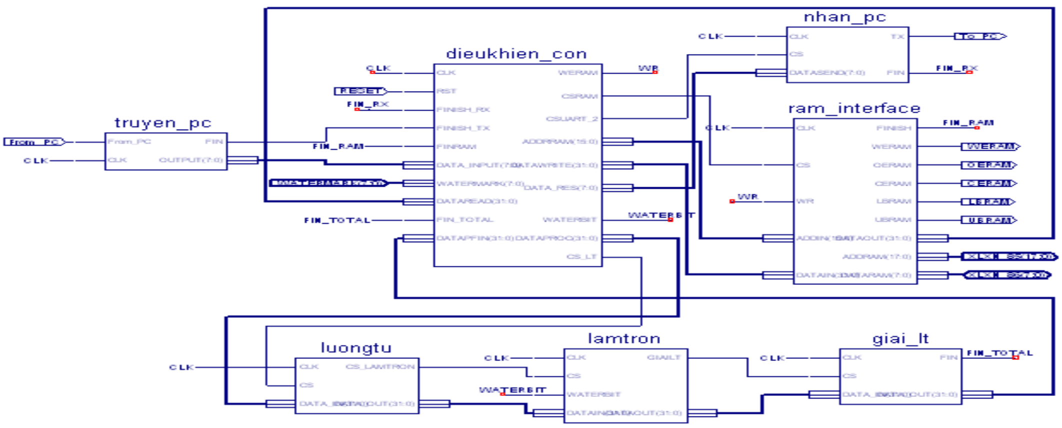


Figure 12 **Detail block diagram of LSB implementation on FPGA**