International Journal of Enhanced Research in Science Technology & Engineering, ISSN: 2319-7463
Vol. 4 Issue 6, June-2015, pp: (37-43), Impact Factor: 1.252, Available online at: www.erpublications.com

FPGA based Audio Watermarking using Empirical Mode Decomposition

Piyush S. Jain[1], Dr. Suresh N. Mali[2]
[1,2]Sinhgad Institute of Technology and Science, Narhe, Pune-41, India

Abstract: Digital watermarking is the technique of inserting specific information into signal, data, image or video. Copyright protection, data authentication, covert communication and content identification can be achieved by Digital watermarking. This work aims at developing a robust and secure blind watermarking technique for speech signal. The speech is watermarked using Virtex-5 FPGA which implements a Micro Blaze softcore processor. The watermarking is done using EMD algorithm in MATLAB which is converted to C/C++ code to be used in FPGA through MATLAB Coder and Xilinx EDK software. The developed hardware finds use in public safety digital radio communications and such an interoperable emergency communication is integral to initial response, health of the masses, public safety, security of the nation and economic stability.

Keywords: Empirical mode decomposition, intrinsic mode function, audio watermarking, FPGA, microblaze, virtex-5, xilinx, matlab.

I. INTRODUCTION

Due to rapid progress in wireless communication systems, extreme prevalence mobile systems, and smart card technology, information is more vulnerable to abuse. For these reasons, it is important to make information systems secure to protect data and resources from malicious acts. Digital watermarking is an effective solution for protection of multimedia data. Watermarking means hiding of secret data that is robust and imperceptible within the host data along with a proper method for watermark detection. Digital watermarking has been proposed for a variety of applications, including content protection, authentication, and digital rights management. The current systems used for public safety radio are employed by police, fire department and other emergency services. These radios have a single encryption and are transmitted over open channels. Hence they need a more robust and secure method for checking the authenticity of the received signal. Moreover, open channel radio is highly vulnerable to attacks. To test whether the signal is genuine or not, a blind watermarking scheme should be very useful. Hence this project aims at developing digital watermarking hardware for security and authorization purposes. Many works have been reported for audio watermarking based on discrete wavelet transform (DWT) method that claim increased bit rate, increased SNR, minimum audio-cover period, quality of the watermarked audio and watermark robustness against audio attacks as seen in [2] and [3].

A new method of first decomposing the signal by singular value decomposition and then applying watermark has the advantage of high improvement in audio-cover period is demonstrated in [4]. DCT method has been used successfully in large capacity digital audio watermarking also employing DWT as reported in [5]. Various other attempts of audio watermarking using DCT scheme have been effective with different advantages like in [6] is tested for additive noise and cropping and method in [7] show good robustness to resample and MP3 compression attacks. A blind watermarking method using DCT is presented in [8]. This paper deals with signal processing through FPGA hardware. The target hardware is the Genesys circuit board based on a Xilinx Virtex-5 LX50T which is a well-equipped and high-end digital circuit development platform. The large on board collection of high-end peripherals, including Gbit Ethernet, HDMI Video, 64-bit DDR2 memory array, and audio and USB ports make the Genesys board an ideal host for complete digital systems, including embedded processor designs based on Xilinx's MicroBlaze. Genesys is compatible with all Xilinx CAD tools, including ChipScope, EDK, and the free WebPack, so designs can be completed at no extra cost. The signal processing and watermarking is done in MATLAB. The MATLAB code is later converted to C/C++ code through MATLAB Coder tool. A FPGA consists of reconfigurable logic blocks and reconfigurable interconnects. These reconfigurable logic blocks and interconnects can be made to behave like a processor. Such a processor is called a soft core processor because it is not actually present but the FPGA is behaving like one. Xilinx FPGA and software tools support the design and implementation

of the Microblaze soft core processor on the FPGA. The C/C++ code generated is used to program the MicroBlaze soft core processor in the Virtex-5 FPGA using Xilinx ISE Design Suite software and its allied utilities. The results of watermarking can be verified using MATLAB. Also the watermarked audio signal can be tested against various attacks in MATLAB. This complete flow of using these software tools is shown in figure 1.

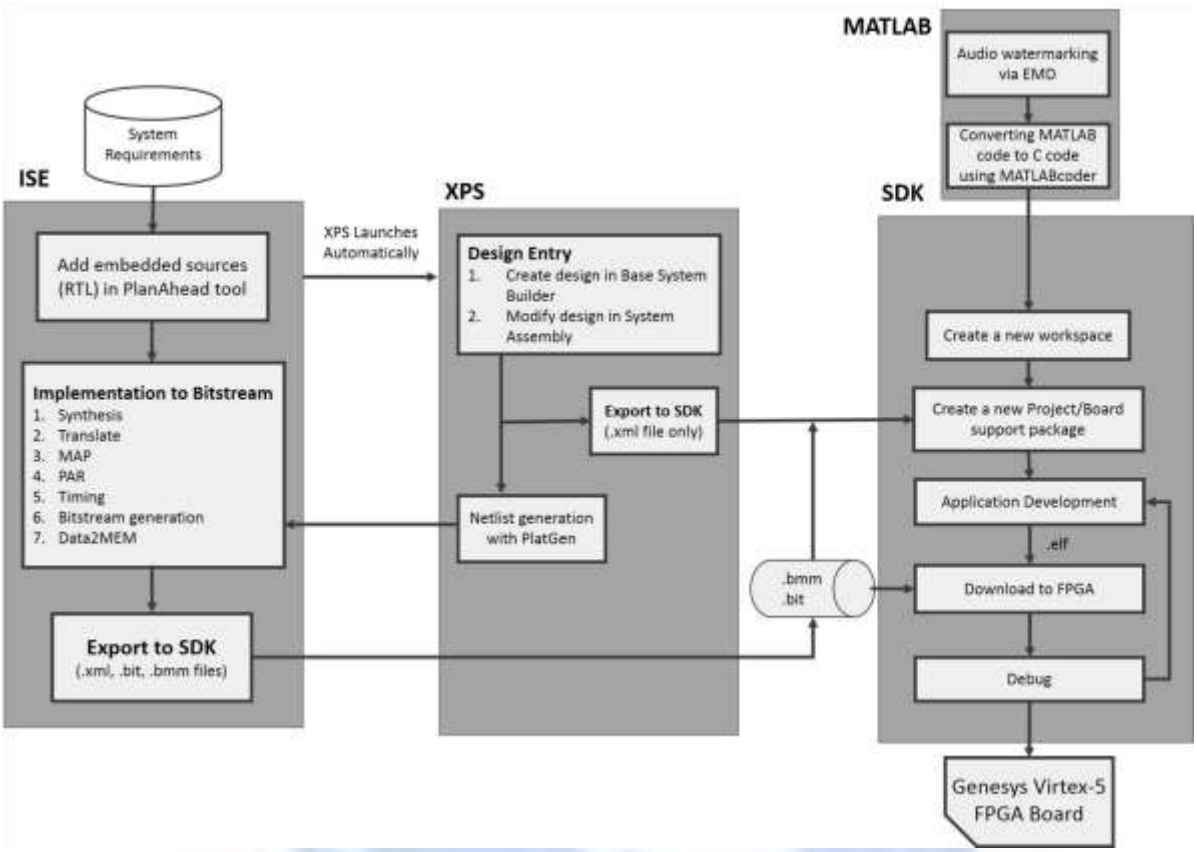

Fig 1. Workflow through various software tools

II. PROPOSED AUDIO WATERMARKING METHODOLOGY

The idea of the proposed watermarking method is to hide into the original audio signal a watermark together with a Synchronized Code (SC) in the time domain. The input signal first undergoes segmentation into frames and each frame is further decomposed into IMFs using EMD. The data sequence, that is to be embedded in the extrema of a set of consecutive last-IMFs consists of informative watermark bits sandwiched by SCs, is a binary one. For each extrema a bit (0 or 1) is inserted. There is a variation of the total number of bits that will be embedded in the last-IMF of each frame as the number of IMFs and their extrema are dependent on the amount of data in each frame. The extrema of last IMF of only one frame is not the location of the entire binary sequence to be embedded. The length of the Watermark and SCs is comparatively much larger than number of extrema in the last-IMF of a frame). This also depends on the length of the frame. The figure 3 shows the watermark embedding procedure. The length of binary sequence to be embedded is equal to $2N_1 + N_2$, where N_1 and N_2 are the numbers of bits of SC and watermark respectively. Thus, these $2N_1 + N_2$ bits are spread out on several last-IMFs (extrema) of the consecutive frames. Further, this sequence of $2N_1 + N_2$ bits is embedded P times. Finally, inverse transformation (EMD^{-1}) is applied to the modified extrema to recover the watermarked audio signal by superposition of the IMFs of each frame followed by the concatenation of the frames. For data extraction, the watermarked audio signal is split into frames and EMD applied to each frame. Binary data sequences are extracted from each last IMF by searching for SCs.
EMD being fully data adaptive, thus it is important to guarantee that the number of IMFs will be same before and after embedding the watermark. In fact, if the numbers of IMFs are different, there is no guarantee that the last IMF always contains the watermark information to be extracted. To overcome this problem, the sifting of the watermarked signal is forced to extract the same number of IMFs as before watermarking. The proposed watermarking scheme is blind, that is, the host signal is not required for watermark extraction (Figure 3). Audio watermarking is done using MATLAB in which the audio file of .wav format is taken as input and a watermark of binary image in .png format is applied to it. The watermarked audio is saved bearing the name ‘watermarkedaudio’ in .wav format.

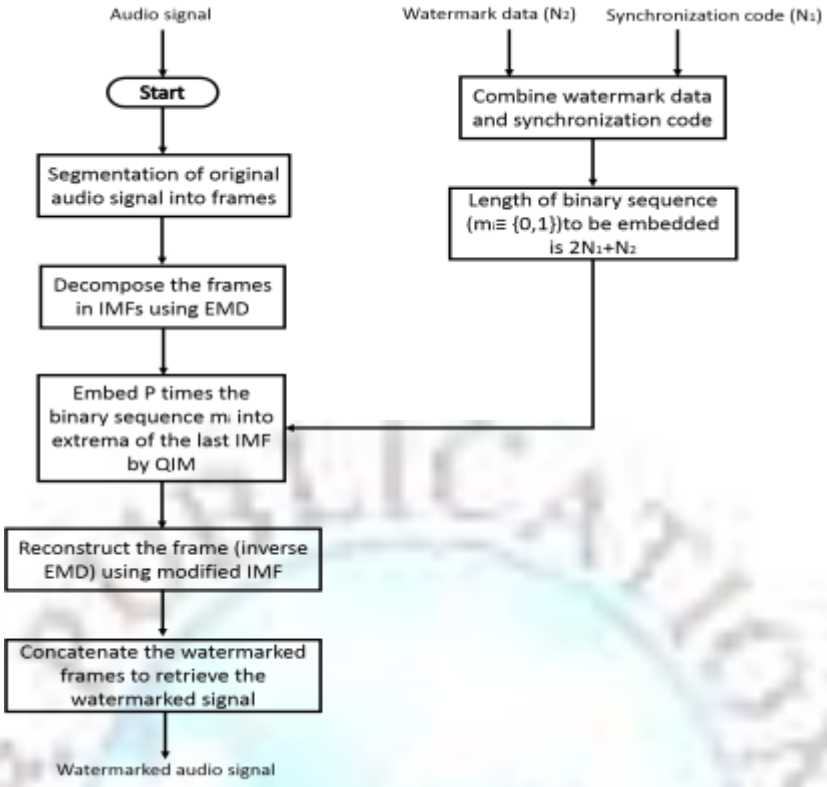

Fig. 2. Watermark Embedding

The steps undertaken for watermark embedding and extraction are given below.

A. SYNCHRONIZATION CODE

A SC is used to find the position of embedding the hidden watermark data in the host signal. Cropping and shifting attacks do not affected this code [1]. Let X be the original SC and Y be an unknown sequence of the same length. Sequence V is considered as a SC if only the number of different bits between U and V, when compared bit by bit, is less or equal than to a predefined threshold τ.

B. WATERMARK EMBEDDING

Before embedding, SCs are combined with watermark bits to form a binary sequence denoted by $m_i \in \{0,1\}$, i-th bit of watermark. Basics of our watermark embedding are shown in figure 2 and detailed as follows:
Using QIM in [10]

$$e_i^* = \lfloor e_1/S \rfloor \cdot S + sgn\left(\frac{S}{4}\right) \; if \; m_i = 0$$

$$e_i^* = \lfloor e_1/S \rfloor \cdot S + sgn\left(\frac{3S}{4}\right) \; if \; m_i = 1$$

Where ei and ei* are the extrema of the host audio signal and the watermarked signal respectively. Sgn function is equal to "+" if ei is a maxima, and "-" if ei is a minima. $\lfloor \cdot \rfloor$ denotes a floor function and S denotes the embedding strength chosen to maintain the inaudibility constraint.

Fig 3 below shows the last IMF of an audio frame before and after watermarking.

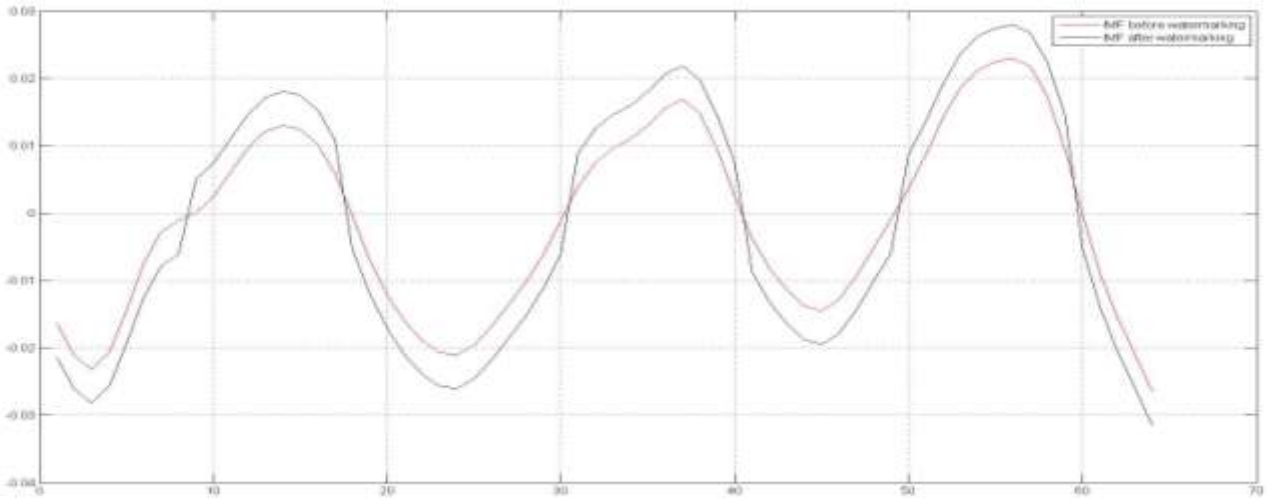

FIG 3. LAST IMF OF AN AUDIO FRAME BEFORE AND AFTER WATERMARKING

C. WATERMARK EXTRACTION

For watermark extraction, host signal is split into frames and EMD is performed on each one as in embedding. The binary data is extracted using rule given below. Then the extracted data is searched for SCs. Watermarking embedding is summarized in Figure 2 and the binary watermark is shown in Figure 4.

$$m_i^* = 1 \text{ if } e_i^* - \left\lfloor \frac{e_1}{S} \right\rfloor \geq \text{sgn}(S/2)$$
$$m_i^* = 0 \text{ if } e_i^* - \left\lfloor \frac{e_1}{S} \right\rfloor < \text{sgn}(S/2)$$

This procedure is repeated by shifting the selected segment (window) one sample at time until a SC is found. With the position of SC determined, we can then extract the hidden information bits, which follows the SC. Let $y = \{m_i\}$ denote the binary data to be extracted and denote the original SC. To locate the embedded watermark we search the SCs in the sequence $\{m^*_i\}$ bit by bit. The extraction is performed without using the original audio signal.

Fig 4. Binary watermark

III. RESULTS OF AUDIO WATERMARKING

The audio watermarking algorithm is developed in MATLAB and tested for various attacks. The performance of this method is evaluated in terms of Bit Error Rate (BER). According to International Federation of the Photographic Industry (IFPI) recommendations, a watermark audio signal should maintain more than 20 dB SNR. To evaluate the watermark detection accuracy after attacks, we used the BER defined as follows:

$$BER(W, \hat{W}) = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} W(i,j) \oplus \hat{W}(i,j)}{M \times N}$$

where is $\oplus$ the XOR operator and $M \times N$ are the binary watermark image sizes. W and W' are the original and the recovered watermark respectively. BER is used to evaluate the watermark detection accuracy after signal processing operations.

To assess the robustness of our approach, different attacks are performed:

- Noise: White Gaussian Noise (WGN) is added to the watermarked signal until the resulting signal has an SNR of 20 dB. Even Pink noise was added and BER was calculated.
- Filtering: Filter the watermarked audio signal using Wiener filter.
- Cropping: Segments of 512 samples are removed from the watermarked signal at thirteen positions and subsequently replaced by segments of the watermarked signal contaminated with WGN.
- Resampling: The watermarked signal, originally sampled at 44.1 kHz, is re-sampled at 22.05 kHz and restored back by sampling again at 44.1 kHz.

Table I shows various attacks and the percentage bit error rates calculated. Also figure 5 shows the histograms of original and watermarked version of the audio signal. It is important to note that here histograms align perfectly with each other show the inaudibility of the watermark. The watermarked version's histogram has a slight increase in the central region indicating that the watermark data is embedded in the extrema of last IMFs of the input signal.

TABLE I: BER OF EXTRACTED WATERMARK FOR AUDIO SIGNAL BY PROPOSED APPROACH

Attack Type	% BER
No Attack	2.2461
Gaussian Noise	10.0586
Filtering	14.1602
Cropping	4.9805
Resampling	2.2461
Pink Noise	11.2305

IV. RESULTS OF FPGA IMPLEMENTATION

The developed system for audio watermarking is synthesized and configured in Xilinx ISE Design Suite. As can be seen in Table II, the main consumed FPGA resources are logic components; in addition, the hardware resources in FPGA are adequate. The software platform Xilinx ISE Design Suite automatically utilizes the logic components to realize all components needed by soft core processor. Notably, the slice register utilisation is around 20%, which indicates that more complex circuitry can be implemented in the remaining resources of the FPGA.

The Genesys board includes a National Semiconductor LM4550 AC 97 audio codec (IC19) with four 1/8 audio jacks for line-out (J16), headphone-out (J18), line-in (J15) and microphone-in (J17). Audio data at up to 18 bits and 48- kHz sampling is supported, and the audio in (record) and audio out (playback) sampling rates can be different. The microphone jack is mono; all other jacks are stereo. The headphone jack is driven by the audio codec's internal 50mW amplifier. The input given to this board at the time of testing was from a laptop through its line-in jack. The audio watermarking performance was good with no audible distortions observed during playback. This audio is recorded back into the computer to verify the watermarking operation via extraction algorithm implemented in MATLAB. The results of this extraction are found to be consistent with those of software implementation.

Table II: Resource Utilization In Fpga

Slice Logic Utilization	Used	Available	Utilization
Number of Slice Registers	6,298	28,800	21 %
Number of Slice LUTs	5,399	28,800	18 %
Number used as logic	5,080	28,800	17 %
Number used as Memory	307	7,680	3 %
Number of occupied Slices	2,926	7,200	40 %
Number of LUT Flip Flop pairs used	8,445		
Number with an unused Flip Flop	2,147	8,445	25 %
Number with an unused LUT	3,046	8,445	36 %
Number of fully used LUT-FF pairs	3,252	8,445	38%
Number of slice register sites lost	1,231	28,800	4 %
Number of bonded IOBs	148	480	30 %
Number of LOCed IOBs	148	148	100 %
IOB Flip Flops	295		
Number of BlockRAM/FIFO	25	60	41 %
Total Memory used (KB)	900	2,160	41 %
Number of BUFG/BUFGCTRLs	6	32	18%
Number of IDELAYCTRLs	3	16	18 %
Number of BSCANs	1	4	25 %
Number of BUFIOs	8	56	14 %
Number of DSP48Es	3	48	6 %
Number of PLL ADVs	1	6	16 %

V. CONCLUSION

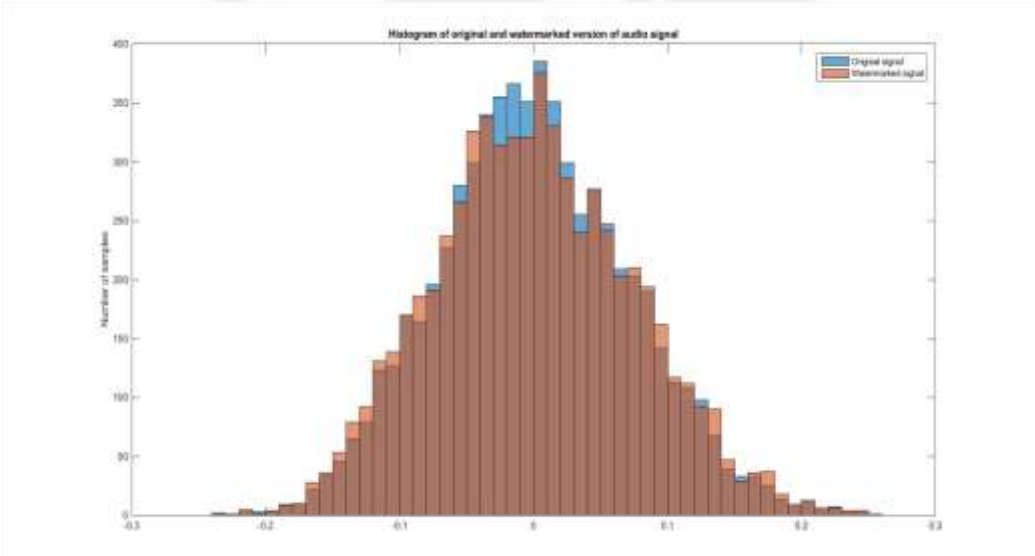

Fig 5. Histogram of audio signal and its watermarked version

In this paper a new adaptive watermarking scheme based on the EMD is implemented on FPGA hardware. Watermark is embedded in very low frequency mode (last IMF), thus achieving good performance against various attacks. Watermark is associated with synchronization codes and thus the synchronized watermark has the ability to resist shifting and cropping. Data bits of the synchronized watermark are embedded in the extrema of the last IMF of the audio signal based on QIM. Extensive simulations in software and testing on hardware over different audio signals indicate that the proposed watermarking scheme has good robustness against common attacks as compared to other recently proposed algorithms. In all audio test signals, the watermark introduced no audible distortion. Experiments demonstrate that the watermarked audio signals are indistinguishable from original ones. These performances take advantage of the self-adaptive decomposition of the audio signal provided by the EMD. This method is a blind watermarking technique involving easy calculations and doesn't need the original audio signal for recovery of the watermark data. In the conducted experiments the embedding strength is kept constant for all audio files. Moreover, this hardware implementation also points to the fact that this method is suitable for real-time implementations and can be improved by using the dedicated DSP resources of the Virtex-5 FPGA.

REFERENCES

[1]. Kais Khaldi and Abdel-Ouahab Boudraa, "Audio Watermarking Via EMD", IEEE Transactions on Audio, Speech, And Language Processing, Vol. 21, No. 3, March 2013.

[2]. Alshammas H.A., "Robust audio watermarking based on dynamic DWT with error correction", Proceedings of ITU Kaleidoscope: Building Sustainable Communities (K-2013), April 2013.

[3]. Elshazly A.R, Fouad M.M., "Secure and robust high quality DWT domain audio watermarking algorithm with binary image," Proceedings of Seventh International Conference on Computer Engineering Systems (ICCES), Nov. 2012.

[4]. Al-Yaman, Al-Taee, "Audio-watermarking based ownership verification system using enhanced DWT-SVD technique," Proceedings of 9th International Multi-Conference on Systems, Signals and Devices (SSD), March 2012.

[5]. Keqiang Ren, Huihuan Li, "Large capacity digital audio watermarking algorithm based on DWT and DCT," International Conference on Mechatronic Science, Electric Engineering and Computer (MEC), Aug. 2011.

[6]. Duymaz E., Akan A., "A novel DCT based digital audio watermarking method," 20th Signal Processing and Communications Applications Conference (SIU), April 2012.

[7]. Xia Zhang, Di Chang, Wanyi Yang, Qian Huang, et. al., "An audio digital watermarking algorithm transmitted via air channel in double DCT domain," International Conference on Multimedia Technology (ICMT), July 2011

[8]. Maher E., Mohamed, Koubaa, Chokri B.A., "DCT Based blind audio watermarking scheme," IEEE International Conference on Signal Processing and Multimedia Applications (SIGMAP), July 2010.

[9]. N.E.Huang et al., "The empirical mode decomposition and Hilbert spectrum for nonlinear and non-stationary time series analysis," Proc. R. Soc., vol. 454, no. 1971, pp. 903995, 1998.

[10]. B. Chen and G. W. Wornell, "Quantization index modulation methods for digital watermarking and information embedding of multimedia," J. VLSI Signal Process. Syst., vol. 27, pp. 733, 2001.

[11]. "EDK Concepts, Tools, and Techniques: A Hands-On Guide to Effective Embedded System Design UG683," Xilinx Inc., (v14.6) June, 2013.

[12]. Pullman WA, "Adept Software Basic Tutorial," Digilent Inc., February, 2010.

[13]. "Virtex-5 FPGA User Guide," Xilinx Inc., UG190 (v5.4), March 2012.