Chapter

Hardware Implementation of Audio Watermarking Based on DWT Transform

Amit M. Joshi

Abstract

Presently, the duplicate copy of an audio can be generated with great ease using some smart devices, and transmitted over the internet which raises concern over copyright and privacy. Digital audio watermarking is a procedure to insert some data bits known as watermark into audio signal. Then the audio with watermark is to be transmitted to end user or made public. The proposed algorithm is used to insert a binary watermark image into a detailed coefficient of the Daubechies 9/7-based DWT transform. A watermark is dispersed consistently in low frequencies, which builds the robustness and inaudibility of the watermark data. Further, the watermark is embedded into an audio signal to have robust system against audio attacks and inaudible performance. The algorithm is verified using MATLAB and subsequently implemented on FPGA hardware to verify the real-time performance. Hardware implementation helps to embed the watermark at the same instance when audio is being captured. The results show promising application for real-time audio applications.

Keywords: audio, digital watermark, FPGA, real-time, robustness

1. Introduction

In a present digital era, a digital file like audio can be copied easily to a computer and other smart devices, and distributed on open network. However, this has prompted issues such as maintaining copyright, ownership, particular person authentication, privacy, and sensitive information loss [1]. The possible solution is to insert some ownership data bits into the audio which would be extracted for the purpose of the authentication. Digital audio watermarking is a technique where a watermark is embedded in the original audio media file. Subsequently, the secured watermarked may be transmitted over internet to any other person. Inaudibility and robustness are two primary characteristics of a digital audio watermark. Robustness is defined as the ability of watermark to resist channel attacks like echo addition, filtering and Gaussian noise, etc. [2]. Inaudibility means the insertion of the watermark should not have any impact on final watermarked audio. Ownership protection helps to identify the content for the originator in order to protect his copyrights. Illegal use of audio without consent, leaking sensitive information, etc. can be prohibited by embedding owner signature into original audio in real-time

[3]. The main objective is to design an algorithm which is robust, blind and inaudible and useful for audio applications.

1.1 Digital watermarking overview

The following section gives a brief overview of digital watermarking. Some basic terms, watermark classifications, watermark properties, and applications covered under this chapter. The following list contains the meaning of some standard terms used in this chapter.

- Host audio is the source audio signal.
- Watermark is defined as a signal consisting of data embedded into a host/carrier audio signal.
- Watermark Embedding is the process of inserting the ownership data into host audio.
- Blind Watermarking is a technique in which there is no need of source audio for watermark extraction.
- Watermark extraction is a procedure to retrieve back to our embed watermark.
- The payload is the size of the message encoded in object [4].

1.2 Problem statement

There are so many audio watermarking algorithms which are implemented in previous year. Most of the algorithms are implemented on the MATLAB only and then it checks its robustness and inaudibility. In MATLAB, the transform function is generally used and according to that an audio watermarking algorithm is applied to frequency domain. In MATLAB, the transform function is generally used and according to that an audio watermarking algorithm is applied in the frequency domain. The power consumption is also unknown and also do not have any knowledge about execution time of the algorithm. These are the some fundamental requirement to design any algorithm on hardware so MATLAB does not provide any kind of hardware support. The hardware implementation of algorithms are achieved on DSP processor and also on GPU processor level. DSP processor and GPU processor may give hardware implementation but its hardware complexity is very high and they are not compatible with the real-time applications [5]. So, VLSI architecture is the best suitable platform for reducing hardware complexity and designing on real-time applications.

1.3 Objectives

The Proposed design of audio watermarking algorithm is implemented on MATLAB. Subsequently VLSI architecture of the audio watermarking algorithm is developed. Then a Forward DWT transform algorithm is developed in Xilinx ISE which is followed by inverse DWT algorithm. Then design VLSI architecture of blind audio watermarking algorithm is developed. Here the main objectives of this proposed work is design VLSI architecture of the blind audio watermarking algorithm and also check its area and timing calculation. The proposed work is also designed to have compatibility with real-time application.

1.4 Previous work and my contribution

Digital audio watermarking is used for correct owner identification, prevention of fragile and copying and also providing a particular person authentication of their digital property. There are many digital audio watermarking algorithms are designed and simulated on MATLAB platform. So many types of audio watermarking methods present in a previous year [6–8]. Also, there is a DWT SVD-based audio watermarking algorithm is implemented in previous work [9]. This work based on semi-blind audio watermarking-based algorithm and a digital watermark is applied on DWT-SVD transform with robustness and imperceptible. The proposed algorithm is a blind digital audio watermarking scheme using DWT algorithm. There are several hardware implementation of the DWT algorithm [10–12]. In the proposed algorithm, the reduced the complexity of the DWT is designed along with its inverse DWT algorithm. The real-time application requires high speed of the algorithm. Our algorithm gives less delay with complete synchronization which does not require any control segment as suggested by many scholars which increase delay. Here, the hardware implementation uses only adders, subtractors and shifters so multiplier-less designed would help to have the hardware efficient and very fast algorithm.

1.5 Hardware solution

The scheme of watermarking is implemented either using software or hardware. In a software implementation, a watermark algorithm is executed on a processor. The software implementation is flexible, but the software implementation is used to embed watermark on offline process where algorithm runs on PC for audio captured through the device. However, the hardware implementation helps to insert the watermark online when the audio is being recorded itself. Then again, in a hardware implementation, a watermark calculation is entirely performed in specially crafted hardware. A hardware implementation consumes less area and less power contrasted with a software implementation [5]. The hardware implementation may have the advantage of parallel processing and poses lesser delay compared to software. This chapter is targeting a real time application, so hardware solution is best recommended. Initially, the proposed audio watermarking algorithm is implemented on the MATLAB; however MATLAB provides only the simulation platform to validate the performance [13]. The real-time implementation of the proposed audio watermarking is achieved in Xilinx ISE software and simulated result of the audio watermarking is discussed. Here DWT transform is implemented by using adder/subtractor and shifter only. Then steps of both embedded and extraction process of the digital audio watermarking is implemented. Subsequently, the proposed watermarking is also synthesized using Xilinx ISE14.7.

2. Digital watermarking

Watermarking is a method through which the protected data conveyed without much observable change in the watermarked content. The watermarking process comprises of two main steps: (i) embedding method and (ii) extraction method. The secret key could be used for additional level of security. There are fundamentally three sorts of watermarking methods and are described here: (1) non-blind watermarking, (2) semi-blind and (3) blind watermarking. The process of watermarking that uses the original sound signal during the extraction procedure termed as "non-blind watermarking". The watermarking system uses a portion of the segment or some a part of the input audio signal then it is term as a "semi-blind watermarking". The watermarking system helps to retrieve the watermark without use of original audio signal or a part of an audio signal for extraction process termed as "blind watermarking" [14]. The paper covers proposed novel blind audio watermarking scheme and its hardware implementation is performed in Xilinx ISE. The steps of algorithms are covered in Section 3. The watermark consists of a data sequence of binary bits which is inserted into the host signal. The audio watermarking scheme should have following basic characteristics: inaudibility, payload and robustness. **Figure 1** gives the visual representation of the requirements of data watermarking concept in digital audio, these three requirements forms the corners of the magic triangle.

1. **Inaudibility:** The inserted data has to be "inaudible" in the watermarked digital music. Evaluation of the same is quantified using signal-to-noise ratio (SNR).

2. **Security:** The algorithm should be secure where authorized person should able to only retrieve the watermark. The attempt of extracting watermark is to be unsuccessful for an unauthorized user in any case.

3. **Robustness:** The watermark should not be eliminated or removed by applying common processing techniques such as cropping, nonlinear and/or linear filter, lossy compression, etc.

4. **Paylod (capacity):** It is defined as total information to be embedded in the host without having of any distortion. It is usually defined as the bit rate for the audio signal which is the actual number of bits inserted in the original audio and is provided by bits per second (bps).

5. **Real-time processing:** The process of inserting into the original signal without much delay. It should be able to insert the watermark at same instance when the audio is being recorded.

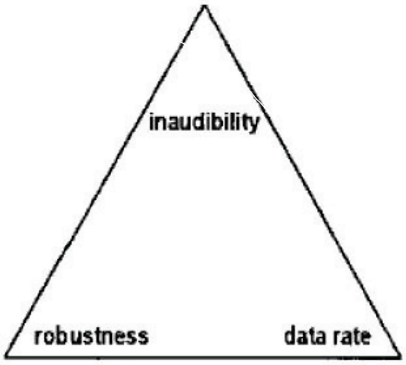


Figure 1.
Performance parameter magic triangle.

3. Proposed audio watermarking

The proposed audio watermarking scheme is blind and robust and is based on DWT transformation. In the proposed scheme, only eight-audio samples of a single frame from two channels, is considered for watermarking process. The details of the embedded and extracted process are shown in the following section.

3.1 Embedding process of DWT-based audio watermarking

Input: original audio, watermark; output: watermarked audio.

The steps of embedding process are as follows:

Step 1: The original audio signal of 16 samples is considered from two channels for further watermark embedding process.

Step 2: DWT transform is applied to obtain an approximate and detailed coefficient of the both channels. Here the approximate and detailed coefficients are low and high pass filter component of the original input signal.

Step 3: Then, binary bits are embedded in the detailed component of an input audio signal. If watermark bit is one then according to

$$P1^{'} = P1 + (I^{*}P1) \text{ in the first channel} \qquad (1)$$

where P_1' = detailed component after watermarking, P_1 = detailed component before watermarking, I = intensity factor and if watermark bit is 0, then 2nd channel detailed component is changed with the first channel detailed component.

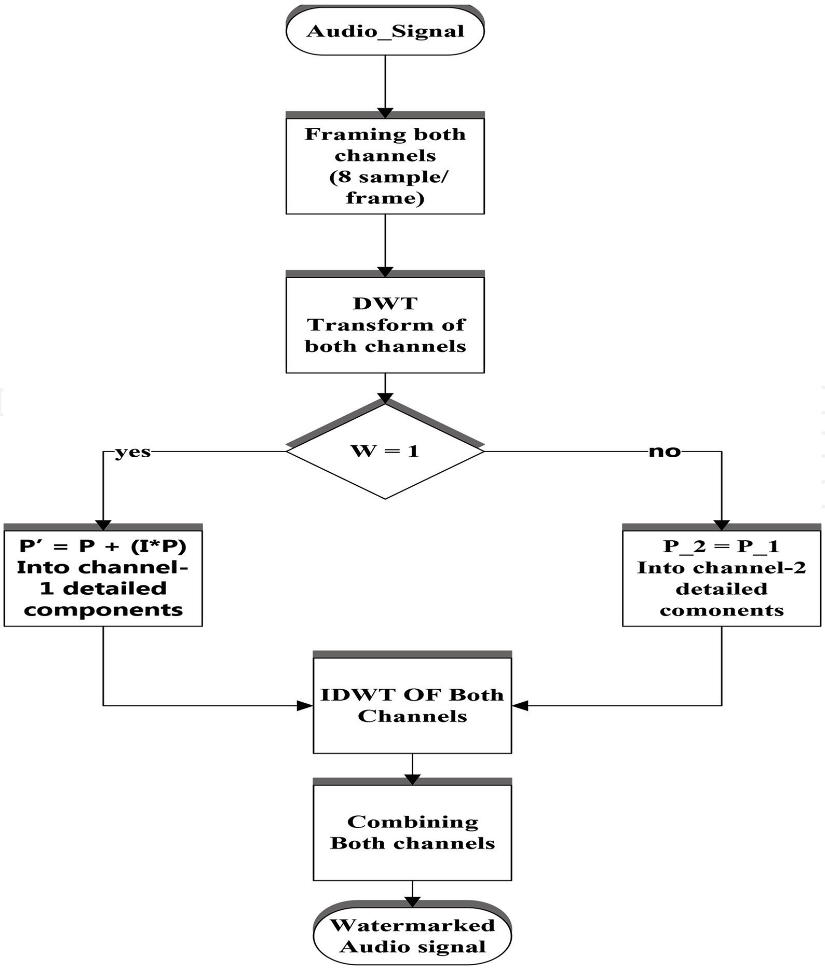

Audio_Signal
Framing both channels (8 sample/ frame)
DWT Transform of both channels
W = 1
yes
no
P′ = P + (I*P) Into channel-1 detailed components
P_2 = P_1 Into channel-2 detailed comonents
IDWT OF Both Channels
Combining Both channels
Watermarked Audio signal

Figure 2.
Flowchart of the embedded process.

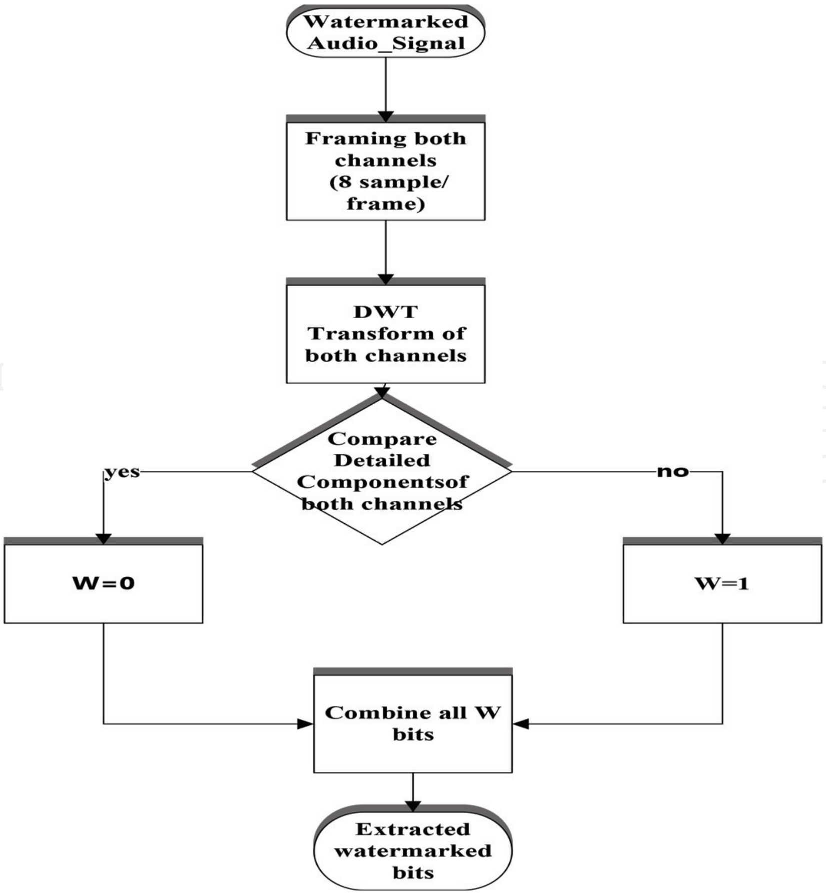

Watermarked Audio_Signal
Framing both channels (8 sample/ frame)
DWT Transform of both channels
Compare Detailed Componentsof both channels
yes
no
W=0
W=1
Combine all W bits
Extracted watermarked bits

Figure 3.
Flowchart of the extraction process.

The flow chart of the embedded process is defined in **Figure 2** of the audio watermarking.

Step 4: After the completion of the embedding process at both channels, the inverse DWT transform is applied in both channels to get watermarked audio signal.

3.2 Extraction process of DWT-based audio watermarking

Input: watermarked audio signal; output: watermark

Step 1: Total 16 samples of both channels of the watermarked audio signal as an input is collected with similar steps followed in embedding process.

Step 2: DWT transform is obtained an approximate and detailed components of the both channel.

Step 3: Now the detailed part of the both channels are observed if they are same then our watermarked bit is 0 otherwise it is 1. The flowchart of the extraction process of the audio watermarking is shown in **Figure 3**.

Step 4: All the watermarked bit into single output to obtain the watermark which was embedded into an audio signal.

4. Hardware implementation of proposed audio watermarking

The architecture of watermark embedding process is defined in **Figure 4**. The process comprises of mainly three modules: DWT module, watermark embedding module and inverse DWT module. Initially, the audio samples are stored in Block RAM1 for the processing. The original watermark is also applied to watermark

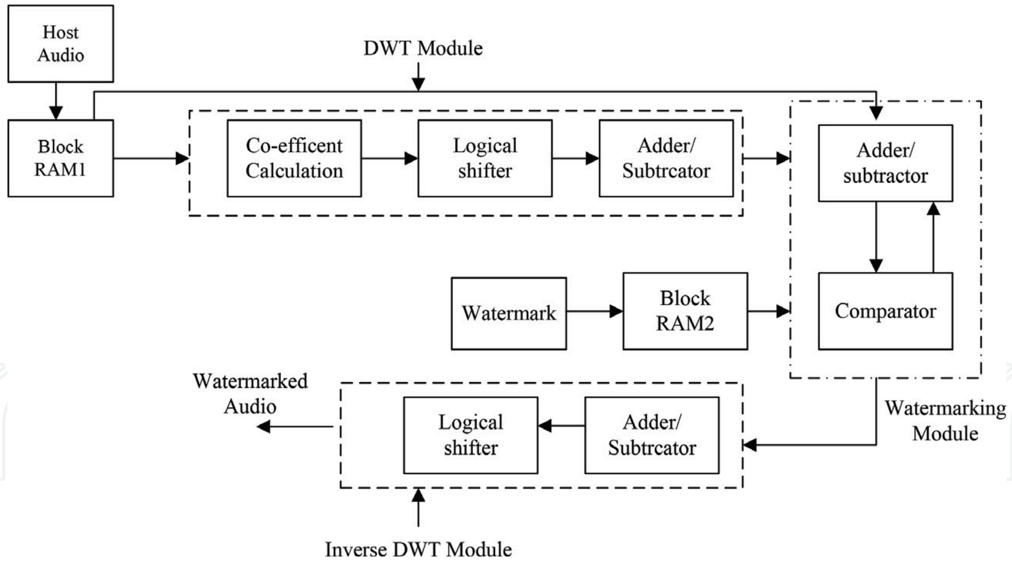


Figure 4.
VLSI architecture of watermark embedding process.

embedding unit. DWT module is used to read the values from the RAM and then converts these values to frequency domain co-efficient.

DWT module has coefficients calculation unit to compute various coefficient for Daubechies filter. After that, these coefficients are processed through watermarking module to insert the watermark. The watermarking module consists of comparator and adder/subtractor to embed the watermark into co-efficient. The comparator is used to take one bit of watermark from block RAM2 and as per bit, the module would decide to embed the watermark of the detailed co-efficient. After the insertion of watermark, the values are fed to inverse integer modules where watermarked audio samples are generated.

4.1 Hardware implementation of DWT

The models for executing of the DWT have mainly grouped in two classifications: (1) convolution filter based [15] and (2) the lifting based [15]. The vital discrete wavelet transform (DWT) is frequently refined by a convolution-based filter implementation using the FIR-filters for doing its transform [16]. FIR filters are applicable for improving the execution of the DWT hardware design [17]. Since a lifting structures have points of interest over a convolution-based regarding computation memory usage and complexity, more consideration is paid to the lifting-based approach. Daubechies and Sweldens [15] proposed the new wavelet scheme by taking into account of the second-generation wavelet. The lifting plan has better performance than convolution filter-based DWT. The lifting plan, which altogether depends on the spatial domain, has numerous favorable circumstances contrasted with filter bank structure, for example, lower complexity and power consumption with relatively reduced area. The lifting-based DWT has fundamental part of high-pass filter and divide the values in low-pass filters where sequence of upper and lower triangular matrices is being formed [18]. The lifting scheme contains mainly three stages, known as, split, predict (P), and update (U). Each of these steps is shown in **Figure 5**. The initial step is separating the original values into odd, and even samples, and then after the odd samples are changed to have the prediction and is obtained as the detail coefficients g_{j+1}. The even value is represented as coarser adaptation of input of the significant portion from determination. The average value of preserved signal, the detailed coefficients would help to revise the

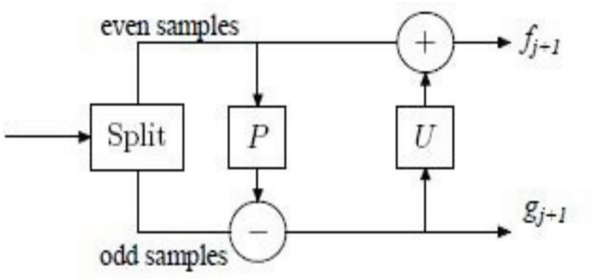


Figure 5.
Forward DWT process.

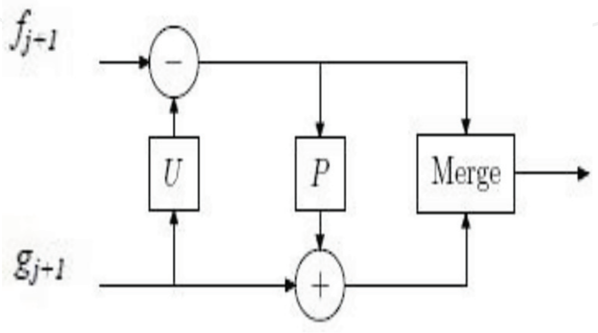


Figure 6.
Inverse DWT process.

even part. The process carried out in update step that creates f_{j+1} approximate coefficients. In order to achieve inverse transform, the sign is going to be exchanged at predict stage and the update stage and all operations are being applied in reversed order as defined in **Figure 6**.

The main objective is to achieve the lower and upper matrices (triangular type) and normalized diagonal matrix by dividing the polyphase matrix of the wavelet filters [19]. As indicated by the fundamental rule, the lifting filter of polyphase matrix of a 9/7 is defined as in Eq. (2).

$$P(z) = \begin{bmatrix} h_e(z) & g_e(z) \\ h_o(z) & g_o(z) \end{bmatrix}$$
$$(\lambda(z)\gamma(z)) = (x_e(z)z^{-1}x_o(z))P(z) \tag{2}$$

where g(z) and h(z) are high pass and low pass filter and is denied as notation of e (even) and o (odd) part respectively. The value is defined in Eq. (3).

$$P(z) = \begin{bmatrix} 1 & \alpha(1+z^{-1}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \beta(1+z) & 1 \end{bmatrix} \begin{bmatrix} 1 & \gamma(1+z^{-1}) \\ 0 & 1 \end{bmatrix}$$
$$\begin{bmatrix} 1 & 0 \\ \delta(1+z) & 1 \end{bmatrix} \begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix} \tag{3}$$

where $\alpha(1+z^{-1})$ and $\gamma(1+z^{-1})$ are the predict polynomials, $\beta(1+z)$ and $\delta(1+z)$ are polynomials which are being updated, and scale normalization factor is denoted as K. And $\alpha = -1.586134342$, $\beta = -0.052980118$, $\gamma = 0.8829110762$, and $\delta = 0.4435068522$ are lifting co-efficient, and K = 1.149604398. For an input x(n) sequence, for n = 0, 1, …, N − 1, the steps of lifting scheme are given as in Eq. (4)

$$\begin{aligned}
s_i^0 &= x_{2n}\\
d_i^0 &= x_{2n+1}\\
d_i^1 &= d_i^0 + \alpha\left(s_i^0 + s_{i+1}^0\right)\\
s_i^1 &= s_i^0 + \beta\left(d_{i-1}^1 + d_i^1\right)\\
d_i^2 &= d_i^1 + \gamma\left(s_i^1 + s_{i+1}^1\right)\\
s_i^2 &= s_i^1 + \delta\left(d_{i-1}^2 + d_i^2\right)\\
d_i &= \frac{d_i^2}{k}\\
s_i &= s_i^2 \times k.
\end{aligned} \tag{4}$$

- Splitting into odd part
- Splitting into even part
- Predict P1
- Update U1
- Predict P2
- Update U2
- Scaling after then obtain detail component
- Scaling after then obtain approximate

Outputs d_i and s_i are low-pass as well as high-pass coefficients of wavelet.

4.1.1 Lifting scheme

Daubechies 9/7-based lifting scheme is shown in **Figure 7**. Each lifting step comprises one update as well as one predict step and that for second time for 2D implementation as P1, P2, and U1, U2, separately.

Pipelined shift-and-add logic plans multipliers used as a part of proposed DWT algorithm. This methodology diminishes the basic way essentially with little increment in latency and area. The shifter, signed adder and signed subtractor for multiplication process is used. For multiplication, alpha, beta, gamma, delta, multiply with K and divide with K module are discussed in **Figure 8**. The values are defined in Eq. (5)

$$\begin{aligned}
\text{where, } |\alpha| &= 1 + \frac{1}{2} + \frac{1}{16} + \frac{1}{32} = 1.59375\\
|\beta| &= \frac{1}{16} = 0.0625\\
|\gamma| &= \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{128} = 0.8828125\\
|\delta| &= \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \frac{1}{128} = 0.4453125\\
|K| &= 1 + \frac{1}{8} + \frac{1}{16} = 1.875\\
\left|\frac{1}{K}\right| &= \frac{1}{2} + \frac{1}{4} + \frac{1}{8} = 0.875
\end{aligned} \tag{5}$$

where, $'S \gg n'$ defines the right shift to n bits, where $|\alpha| \times S = S + (S \gg 1) + (S \gg 4) + (S \gg 5)$. The predict step from first lifting generate odd and

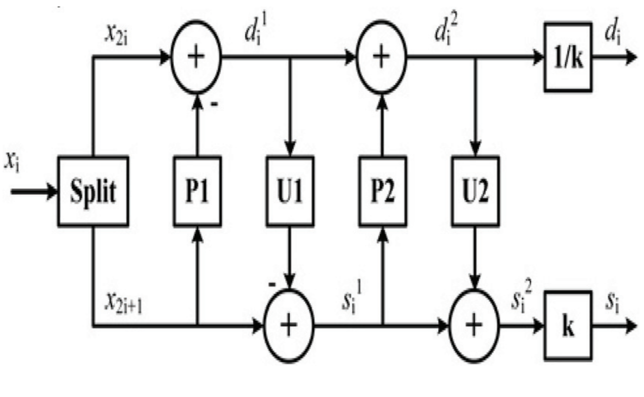
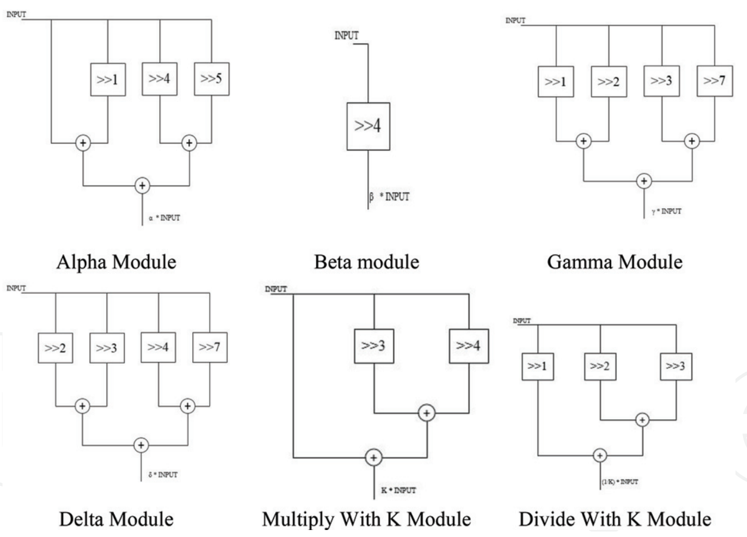


Figure 7.
Lifting plan for Daubechies 9/7 filter.



Alpha Module Beta module Gamma Module

Delta Module Multiply With K Module Divide With K Module

Figure 8.
DWT coefficients calculation.

even contribution after one clock cycle delay. The even value is included with past even input sample (s_i^0, s_{i+1}^0). Then operation of multiplier using primary filter coefficient is done at delay of the second and third clock cycles by applying only shifting and adding operation. After fourth clock cycle, the result of multiplier is considered at odd input sample (d_i^0) to update coefficient (d_i^1). At the end of fifth clock, the present value of predict (d_i^1) and the past value of the predict (d_{i-1}^1) with help of past even info (s_i^0) provides the first value of update (d_i^2). The adders is only the required operation at every clock cycle, thus critical path is defined through an adder delay only. The both phase, predict as well as update, of both stages are implemented in full pipelined approach to increase the speed. The overall lifting implementation comprises of four shifters and seven adders/subtractors. Moreover, the second stage of lifting have overall eight shifters and ten adders. The detail process is defined as in Eq. (6).

For an inverse DWT transform we use alpha, beta, gamma, delta module as same as discussed earlier but we use multiply module with the detailed coefficients and divide module with the approximate coefficients then we go reverse order of all the equation and finally we obtained original audio sample. Total eight samples for DWT transform are considered so after the inverse DWT transform eight samples are obtained. All the inverse DWT transform equation steps are discussed under.

$$\begin{aligned}
s_i^0 &= x_{2n} \\
d_i^0 &= x_{2n+1} \\
d_i^1 &= d_i^0 + \alpha\left(s_i^0 + s_{i+1}^0\right) \\
s_i^1 &= s_i^0 + \beta\left(d_{i-1}^1 + d_i^1\right) \\
d_i^2 &= d_i^1 + \gamma\left(s_i^1 + s_{i+1}^1\right) \\
s_i^2 &= s_i^1 + \delta\left(d_{i-1}^2 + d_i^2\right) \\
d_i &= \frac{d_i^2}{k} \\
s_i &= s_i^2 \times k.
\end{aligned} \tag{6}$$

The proposed design of DWT and inverse DWT would help to have efficient audio watermarking algorithm (**Figure 9**).

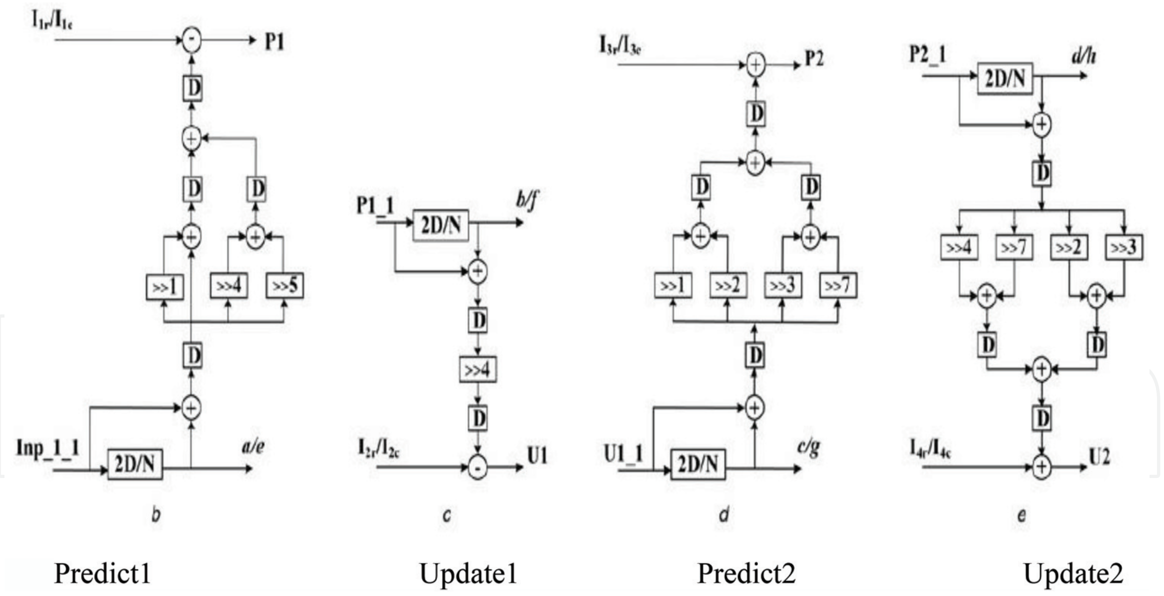


Figure 9.
Predict and update implementation of lifting scheme.

5. Simulation and result

The results are initially developed through MATLAB and then hardware implementation are achieved to verify the real-time implementation.

5.1 MATLAB

Experiments are performed in MATLAB 2010a. The proposed algorithm uses classical/pop music and speech audio clips in order to evaluate the performance [20]. These are three different types of audio clips are considered as they have different characteristics, perceptual properties and energy distribution. These audio signal have various distinct characteristics and also contains some selective features such as low energy, pulse clarity, pitch (in Hz.), inharmonicity, sampling rate (in Hz.), zero crossing rate (per second), spectral irregularity, temporal length (seconds/sample), tempo (in bpm), rms energy, etc. Each audio sample is of mono file of a 16-bit with sampling rate of 44.1 kHz of WAVE format. The watermark is of binary image of a 30×30 bits as in **Figure 10**. The synchronization code is a 16-bit Barker code of value “1111100110101110”. The wavelet is applied with two decomposition levels. Array size m is 50 and the range of quantization step size Δ starts from 0.15 for speech audio and goes up to 0.6 for pop audio signal. The performance of audio watermarking algorithms is quantified by robustness, payload and inaudibility parameter [21].

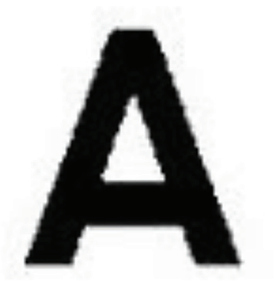


Figure 10.
Watermark.

The inaudibility is measured using signal to noise ratio (SNR). It is a used to calculate the similarity between distorted watermarked audio signal and undistorted original audio signal. SNR is calculated as in Eq. (10):

$$SNR = -10log_{10}\left[\frac{\Sigma_i f_i^2}{(\Sigma_i(\acute{f}_i - f_i)^2)}\right] \tag{7}$$

where f_i is original audio signal, whereas f_i’ is watermarked audio signal. It helps to calculate the noise induced in the watermark and defines the inaudibility.

Robustness: normalized correlation (NC) measure the similarity between original and extracted is given by:

$$NC(w, w') = \frac{\sum_{i=1}^{M}\sum_{j=1}^{M} w(i,j)w'(i,j)}{\sqrt{\sum_{i=1}^{M}\sum_{j=1}^{M} w^2(i,j)}\sqrt{\sum_{i=1}^{M}\sum_{j=1}^{M} w'^2(i,j)}} \tag{8}$$

here w is original watermark, w′ defines the extracted watermark, and i and j are indices to represent the watermark image. Generally, NC is to be considered as equal to 1. The robustness performance is measured using bit error rate (BER) as in Eq. (9).

$$BER(w, w') = \frac{\text{Number of error bits}}{\text{Number of total bits}} \times 100\% \tag{9}$$

The different attacks are considered for the robustness measurement of our proposed algorithm. The detailed of each signal processing attacks are defined and results are defined in **Table 1** [22].

a. **Re-quantization**: original watermarked audio signal of 16 bit/sample is down re-quantized at 8 bits/sample, which further back quantized to 16 bits/sample.

	Pop		Speech		Classical	
Attack	**NC**	**BER**	**NC**	**BER**	**NC**	**BER**
No Attack	1	0	1	0	1	0
Re-quantization	1	0	1	0	1	0
AWGN	0.999	0.003	0.995	0.005	0.999	0.002
Low-pass filter	0.999	0.001	0.997	0.004	0.999	0.001
Re-sampling	1	0	1	0	1	0
Mp3 64 kbps	0.9821	0.041	0.9878	0.037	1	0
MP3 128 kbps	1	0	1	0	1	0
Random cropping	0.997	0.002	0.999	0.001	0.998	0.002
Invert	1	0	1	0	1	0
Echo addition	0.997	0.002	0.998	0.003	0.999	0.002
Denoising	0.996	0.001	0.994	0.005	0.996	0.001
Pitch shifting	0.999	0.001	1	0	1	0

Table 1.
Experimental results for robustness of proposed algorithm.

b. **Additive white Gaussian noise (AWGN)**: to evaluate performance, Gaussian noise is inserted in the watermarked signal till an SNR reaches to 20 db.

c. **Low-pass filtering**: Butterworth filter of second-order is used at 11,025 Hz cutoff frequency.

d. **Re-sampling**: the sampling rate of the watermark signal is 44.1 kHz, further it is re-sampled at 22.05 kHz, and again sampled back at 44.1 kHz.

e. **MP3 compression 64 kbps**: the layer-3 compression of MPEG-1 is being applied. The audio signal with watermark is compressed with 64 kbps bit-rate and subsequently decompressed in the WAVE format.

f. **MP3 compression 128 kbps**: the layer-3 compression of MPEG-1 is being applied. The audio signal with watermark is compressed with 128 kbps bit-rate and subsequently decompressed in the WAVE format.

g. **Random cropping**: total sample of around 10% are cropped at randomly selected positions (front, middle and back).

h. **Invert**: all sample values are inverted in time domain with phase shift 180°.

i. **Echo addition**: an echo signal is added (with a decay of 41% and a delay of 98 ms) inside the watermarked audio signal.

j. **Denoising**: the audio signal with watermark is denoised with function of “automatic click remover” available in Adobe Audition 3.0.

k. **Pitch shifting**: it is most difficult attack for audio watermarking algorithms, because it tends to shift frequency fluctuation. In the results, the pitch is being shifted around one higher degree and one lower degree. These are applied to all three audio signals are shown in given in **Table 1**.

The payload data of the proposed algorithm is shown as:

$$B = {}^{R}/_{m \times 2^{k}} \text{bps} \tag{10}$$

The data payload is considered as 220 bps.

5.2 Comparison with related work

The general comparison is made between our proposed method and two similar methods [23] and is given in **Table 2**. As per reported results in **Table 2**, our proposed algorithm has higher capacity of embedding and lower value of BER. The proposed algorithm may achieve higher performance by reducing payload (which would be achieved by decomposition level increase for wavelet transform or length of array increases). The strength for embedding would increase with that approach.

5.3 Hardware results

The architecture is designed and implemented using Verilog HDL and targeting vertex 5 xc5vlx20t-2ff323 FPGA. We synthesized on Xilinx ISE 14.7. Each input and

Algorithm	Proposed	[24]	[25]
Payload	220	172	196
Noise reduction	0	0	0
BER	20 dB	20 dB	20 dB
Cropping and shifting (robust)	Yes	Yes	No
MP3 64 kbps (BER)	0.041	0.0434	0.01

Table 2.
Comparison with related audio watermarking.

Resources	DWT	Watermark embedding	Inverse DWT
BELs	13,214	5026	10,586
Registers	4946	1258	4268
Adders/subtractor	507	130	468
Multiplier	0	0	0

Table 3.
FPGA report for device utilization.

Resources	Total	DWT	Watermark Embedding	Inverse DWT
Slice	16840	1980 (11%)	570 (3%)	1796 (10%)
Slice FFs	33280	498 (1%)	137 (1%)	412 (1%)
4 input LUTs	33280	3782 (11%)	956 (2%)	3584 (10%)
Bounded IOBs	519	158 (30%)	52 (10%)	135 (26%)
GCLKS	24	1 (4%)	1(4%)	1 (4%)

Table 4.
FPGA report for resource utilization.

Parameter	DWT	Watermark embedding	Inverse DWT
Maximum frequency	36.12 MHz	47.26 MHz	39.32 MHz
Minimum period	27.68 ns	21.15 ns	24.32 ns

Table 5.
Synthesis report for timing analysis.

output is defined using IEEE 754 SP format because of complex calculations during DWT and inverse DWT. **Tables 3** and **4** provides the hardware utilization of targeted FPGA, and **Table 5** has the total computation delay. During synthesis process, the proposed audio watermarking scheme is validated for the real-time performance by FPGA prototyping.

6. Conclusion

The audio watermarking algorithm is proposed for different audio application. The algorithm uses DWT transform during watermarking process. The proposed algorithm has blind detection and has admirable performance against the attacks. A discrete wavelet transform (DWT) represents a data points regarding wavelet at various frequencies. In this chapter, hardware architecture of DWT-based digital audio watermarking is also developed which is used for real-time applications. Digital audio watermarking used in many applications like ownership protections, Tamper detection and localization, and media forensics. Above analysis shows that proposed algorithm gives good SNR with higher inaudibility and NC is also almost equal to 1. Various attacks are considered to check the robustness of the algorithm. This algorithm produces excellent resistance to many attacks. The FPGA prototyping is done for hardware performance measurement for real-time application.

Author details

Amit M. Joshi
National Institute of Technology, Jaipur, Rajasthan, India

*Address all correspondence to: amjoshi.ece@mnit.ac.in

References

[1] Joshi A, Mishra V, Patrikar RM. Real time implementation of digital watermarking algorithm for image and video application. In: Watermarking—Volume 2. Rijeka, Croatia: IntechOpen; 2012

[2] Xiang-yang W, Pan-pana N, Ming-yu L. A robust digital audio watermarking scheme using wavelet moment invariance. The Journal of Systems and Software. 2011;**84**:1408-1421

[3] Hwang M-J, Lee JS, Lee MS, Kang H-G. SVD-based adaptive QIM watermarking on stereo audio signals. IEEE Transactions on Multimedia. 2018; **20**(1):45-54

[4] Joshi AM, Gupta S, Girdhar M, Agarwal P, Sarker R. Combined DWT—DCT-based video watermarking algorithm using Arnold transform technique. In: Proceedings of the International Conference on Data Engineering and Communication Technology; Singapore: Springer; 2017. pp. 455-463

[5] Joshi AM, Mishra V, Patrikar RM. Design of real-time video watermarking based on integer DCT for H. 264 encoder. International Journal of Electronics. 2015;**102**(1):141-155

[6] Hu H-T, Hsu L-Y. A DWT-based rational dither modulation scheme for effective blind audio watermarking. Circuits, Systems, and Signal Processing. 2016;**35**(2):553-572

[7] Hu H-T, Hsu L-Y. Incorporating spectral shaping filtering into DWT-based vector modulation to improve blind audio watermarking. Wireless Personal Communications. 2017;**94**(2): 221-240

[8] Li J-f, Wang H-X, Wu T, Sun X-m, Qian Q. Norm ratio-based audio watermarking scheme in DWT domain. Multimedia Tools and Applications. 2018;**77**(12):1-17

[9] Lalitha NV, Vara Prasad P, UmaMaheshwar Rao S. Performance analysis of DCT and DWT audio watermarking based on SVD. In: 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT); Nagercoil, India: IEEE; 2016. pp. 1-5

[10] Kotteri KA, Barua S, Bell AE, Carletta JE. A comparison of hardware implementations of the biorthogonal 9/7 DWT: Convolution versus lifting. IEEE Transactions on Circuits and Systems II: Express Briefs. 2005;**52**(5):256-260

[11] Algredo-Badillo I, Castillo-Soria FR, Ramirez-Gutierrez KA, Morales-Rosales L, Medina-Santiago A, Feregrino-Uribe C. Lightweight security hardware architecture using DWT and AES algorithms. IEICE Transactions on Information and Systems. 2018;**101**(11): 2754-2761

[12] Goran S, Prokin M, Rajović V, Prokin D. Novel one-dimensional and two-dimensional forward discrete wavelet transform 5/3 filter architectures for efficient hardware implementation. Journal of Real-Time Image Processing. 2016:1-20

[13] Singh R, Mohanty SR, Kishor N, Thakur A. Real-time implementation of signal processing techniques for disturbances detection. IEEE Transactions on Industrial Electronics. 2019;**66**(5):3550-3560

[14] Dragoi I-C, Coltuc D. Adaptive pairing reversible watermarking. IEEE Transactions on Image Processing. 2016; **25**(5):2420-2422

[15] Kłos MJ. Determination of road traffic flow based on 3D Daubechies

wavelet transform of an image sequence. In: International Conference on Computer Vision and Graphics; Cham: Springer; 2016. pp. 573-580

[16] Tiwari VK, Jain SK. Hardware implementation of polyphase-decomposition-based wavelet filters for power system harmonics estimation. IEEE Transactions on Instrumentation and Measurement. 2016;**65**(7):1585-1595

[17] Eminaga Y, Coskun A, Kale I. Hybrid IIR/FIR wavelet filter banks for ECG signal denoising. In: 2018 IEEE Biomedical Circuits and Systems Conference (BioCAS); IEEE; 2018. pp. 1-4

[18] Darji A, Agrawal S, Oza A, Sinha V, Verma A, Merchant SN, et al. Dual-scan parallel flipping architecture for a lifting-based 2-D discrete wavelet transform. IEEE Transactions on Circuits and Systems II: Express Briefs. 2014;**61**(6):433-437

[19] Darji A, Agrawal S, Oza A, Sinha V, Verma A, Merchant SN, et al. Multiplier-less pipeline architecture for lifting-based two-dimensional discrete wavelet transform. IET Computers and Digital Techniques. 2015;**9**(2):113-123

[20] Olanrewaju RF, Khalifa O. Digital audio watermarking: Techniques and applications. In: International Conference on Computer and Communication Engineering (ICCCE 2012); 3–5 July, 2012; Kuala Lumpur, Malaysia; 2012

[21] Al-Haj A. An imperceptible and robust audio watermarking algorithm. EURASIP Journal on Audio, Speech, and Music Processing. 2014;**2014**:37

[22] Karthigaikumar P, Jarline Kirubavathy K, Baskaran K. FPGA-based audio watermarking—covert communication. Microelectronics Journal. 2011;**42**(5):778-784

[23] Ozer H, Sankur B, Memon N. An SVD-based audio watermarking technique. In: Seventh ACM Workshop on Multimedia and Security; 2005. pp. 51-56

[24] Wu S, Huang J, Huang D, Shi YQ. Efficiently self-synchronized audio watermarking for assured audio data transmission. IEEE Transactions on Broadcasting. 2005;**51**(1):69-76

[25] Bhat V, Sengupta I, Das A. A new audio watermarking scheme based on singular value decomposition and quantization. Circuits, Systems, and Signal Processing. 2011;**30**(5):915-927