



FPGA IMPLEMENTATION OF LIFTING SCHEME DAUBECHIES DISCRETE WAVELET TRANSFORM FOR REAL-TIME SIGNAL COMPRESSION

Truong Phong Tuyen

College of Engineering Technology, Can Tho University, Vietnam

ARTICLE INFO

Received date: 07/08/2015

Accepted date: 26/11/2015

KEYWORDS

Altera DE2, DWT, FPGA, speech compression, Verilog HDL

ABSTRACT

In recent years, rapid development of multimedia applications leads to the requirement of huge data storage and high-speed data transfer capacity so that data compression emerges as an attractive topic. However, it is hard to compress sudden-changing signals, especially in high-speed and real-time processing. This paper proposes a hardware solution for real-time signal compression based on Daubechies wavelet transform (Db WT). The set of Db WT factors has been modified into integer number in order to be suitable for hardware implementation. The experiments were conducted on FPGA prototype implementation of real-time speech compression for evaluating the proposed approach.

Cited as: Tuyen, T.P., 2015. FPGA implementation of lifting scheme daubechies discrete wavelet transform for real-time signal compression. Can Tho University Journal of Science. 1: 19-26.

1 INTRODUCTION

Nowadays, data compression techniques are common in communication systems for reducing the size of data. It means that less number of bits can convey the same information as original one. The work aims to save storage space, transmission time, and transmission bandwidth. Compression ratio is used to indicate the efficiency of a compression technique. This ratio is defined as the ratio between the uncompressed and compressed size so that the greater ratio is more efficient in compressing. There are two basic types of compression: lossless and lossy. In lossless compression the original information can be retrieved exactly from the compressed data whereas it cannot be done by lossy because of lacking noninvertible property. Consequently, although lossy can be achieved higher compression ratio than lossless, it is limited to application in which we can tolerate the loss (Wonyong *et al.*, 1997; Rajesh *et al.*, 2001; Abdul *et al.*, 2003; Sun *et al.*, 2013).

Applying transform methods is one of many recommended solutions for data compression. Transform methods convert data from one form to another more convenient to process. The Fourier Transform (FT) is a well-known method for signal analysis. The Short Time Fourier Transform (STFT) is an advanced version of FT that gives a time-frequency presentation of the signal. It can be used to analyze unpredictable (non-stationary) signals. However, it cannot have arbitrarily good time and frequency resolutions. In this case, one must trade one for the other. The Wavelet Transform (WT) therefore was proposed as a new method for time-frequency analysis. Wavelet Transform provides the better resolution in both frequency and time domain at the same time based on multi-resolution technique, which is different frequencies be analyzed at different resolutions. This transform is efficient in analyzing non-stationary and transient signals (Wonyong *et al.*, 1997; Rajesh *et al.*, 2001; Abdul *et al.*, 2003; Hussein *et al.*, 2010).

Furthermore, due to the wavelet decomposition offers both time and frequency domain, it is more efficient than the traditional techniques to not only decompose but also reconstruct the non-stationary signals. The translated and scaled mother wavelets are capable of providing multi-resolution of the non-stationary speech signal through low pass and high pass filter banks (Polikar, 1999). The low pass filtered speech signals keep most of the speech energy and perception. Therefore, after the high-pass filtered components are discarded, and the low pass filtered components are kept, the speech will be compressed (Johnson, 1996). Because of these reasons, lossy compression technique is used in this research.

For using a digital system to process the input signal, it is more convenient to define Discrete Wavelet Transform, the discrete version of the wavelet transform. Besides, the lifting versions of wavelet transform have several advantages. They are memory efficient and do not require a temporary memory array. In addition, the inverse transform is the mirror of the forward transform (Ingrid *et al.*, 1998; James, 1999; Ivan, 2003).

In DWT speech compression, the speech signals after being converted into digital, these samples were compressed. Three-level speech compression was performed in real time with the compression ratio 8:1. This compression data were written into SDRAM memory. To reconstruct these signals, the data can be read out from the memory, then up sampled by 8 and converted into analog signals. The reconstructed sound could be played clearly in the external speakers without audible background noise.

The initial study of compression algorithm is evaluated using MATLAB. Verilog Hardware Description Language (HDL) is used to implement the hardware architecture of the speech compression system. The Verilog HDL finite state machine (FSM) designs are verified by running ModelSim simulation. Finally, Altera Cyclone II FPGA device on the Altera DE2 development kit is chosen as the prototyping platform.

This remainder of this paper is organized as follows. Section 2 introduces the lifting scheme DWT theory. How to design the lifting scheme Daubechies DWT block on FPGA is then described in section 3. Finally, in section 4 we discuss the results and close by briefly discussing future directions of this work.

2 LIFTING SCHEME DAUBECHIES DISCRETE WAVELET TRANSFORM

Daubechies wavelets are compact orthogonal filter banks, which satisfy the perfect reconstruction condition. In addition, Daubechies wavelets have maximum number of vanishing moments for a given order so that they can be used to provide the good approximation of the original signal 14. For these reasons, the Daubechies 4-tap (Db4) orthogonal filter bank was chosen for this work. The lifting scheme Daubechies 4-tap (Db4) DWT consists of four-step operations: update1, predict1, update2 and normalization 7. The wavelet-lifting scheme is more memory efficient and does not require a temporary storage as the general Daubechies DWT does. Figure 1 shows the lifting scheme Daubechies DWT diagram.

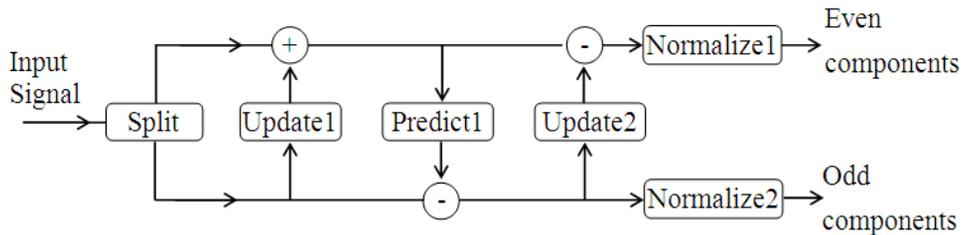


Fig. 1: Lifting scheme Daubechies DWT

Ingrid *et al.* 1998

In Figure 1, the input signals are first splitted into even and odd samples successively, and then update1, predict1, update2, normalization1 and normalization2 stages perform according to equations follows:

Update1: (1)

$$\begin{aligned} Even[n] &= Even[n] + \sqrt{3}Odd[n] \\ &= Even[n] + 1.732Odd[n] \end{aligned}$$

Predict1: (2)

$$\begin{aligned} \text{Odd}[n] &= \text{Odd}[n] - \frac{\sqrt{3}}{4} \text{Even}[n] - \frac{\sqrt{3}-2}{4} \text{Even}[n-1] \\ &= \text{Odd}[n] - 0.433 \text{Even}[n] + 0.067 \text{Even}[n-1] \end{aligned}$$

Update2: (3)

$$\text{Even}[n] = \text{Even}[n] - \text{Odd}[n+1]$$

Normalization1: (4)

$$\text{Even}[n] = \frac{\sqrt{3}-1}{\sqrt{2}} \text{Even}[n] = 0.5176 \text{Even}[n]$$

Normalization2: (5)

$$\text{Odd}[n] = \frac{\sqrt{3}+1}{\sqrt{2}} \text{Odd}[n] = 1.9319 \text{Odd}[n]$$

In this case, Even[n] and Odd[n] denotes for the even and odd samples respectively. Further details about the lifting scheme of Db4 DWT could be found in 7.

3 FPGA DESIGN OF DWT FOR REAL-TIME SPEECH COMPRESSION

In this work, the FPGA design of the lifting scheme Daubechies DWT is proposed that consists of four components: ADC controller, DAC controller, Memory Access Controller (MAC), and the lifting scheme DWT blocks as shown in Figure 2.

In the following section, each block of this design will be described in more details.

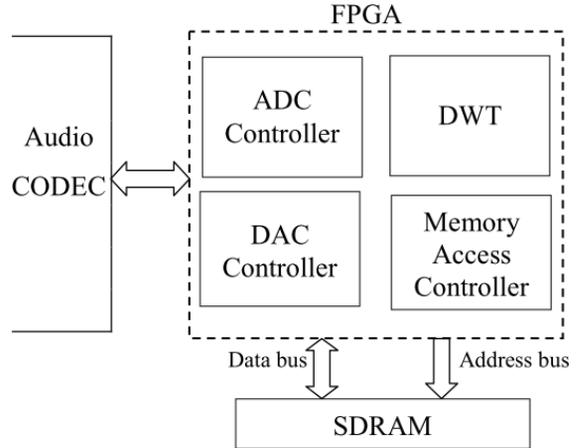


Fig. 2: FPGA design system for speech compression

3.1 Analog to digital controller

Because the output of ADC chip is in serial form, it needs to be converted into the parallel form before storage into SDRAM as references in (Altera Corporation, 2008; Integrated Circuit Solution Inc., 2004). Therefore, the function of Analog to Digital Converter module is to wait to receive serial digitized data from ADC, and then convert to parallel form for writing into SDRAM as well as down-sample the samples. Figure 3 describes Finite State Machine (FSM) of ADC controller.

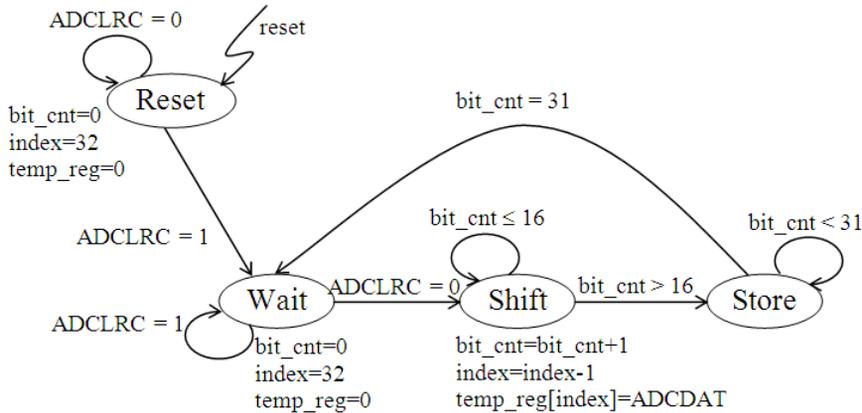


Fig. 3: Finite State Machine of ADC in real time compression

3.2 Digital to analog controller

The function of DAC is to convert reconstructed digital signal back to analog form. The compressed speech read back from SDRAM are appropriately up-sampled, and then sent to DAC for converting

to analog signal to drive external speakers. The FSM of DAC has the same steps as FSM of ADC, but in DAC procedure a flag variable is used to control the down-sampling of output signal. The DAC controller's FSM is illustrated in Figure 4.

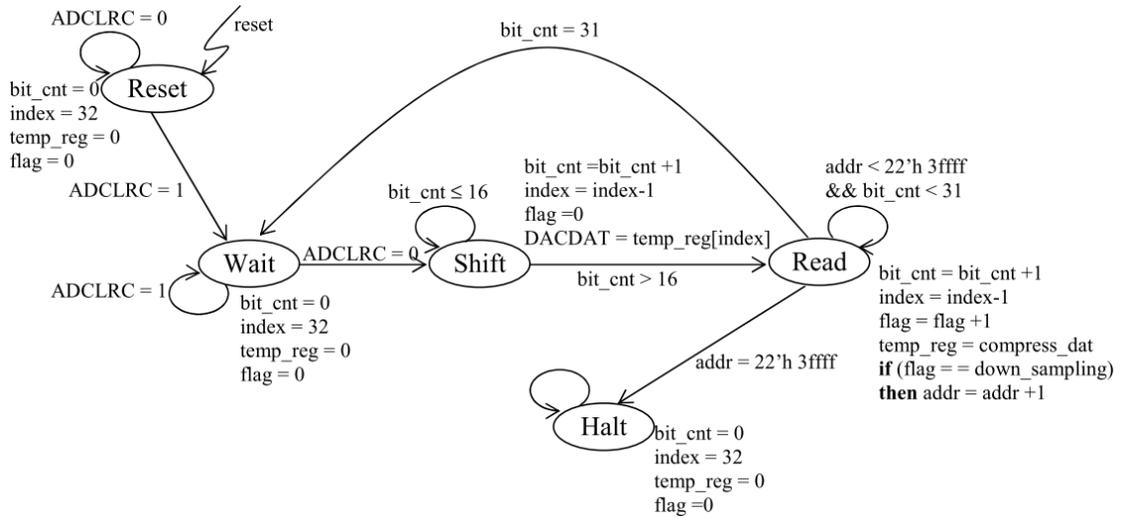


Fig. 4: Finite State Machine of DAC

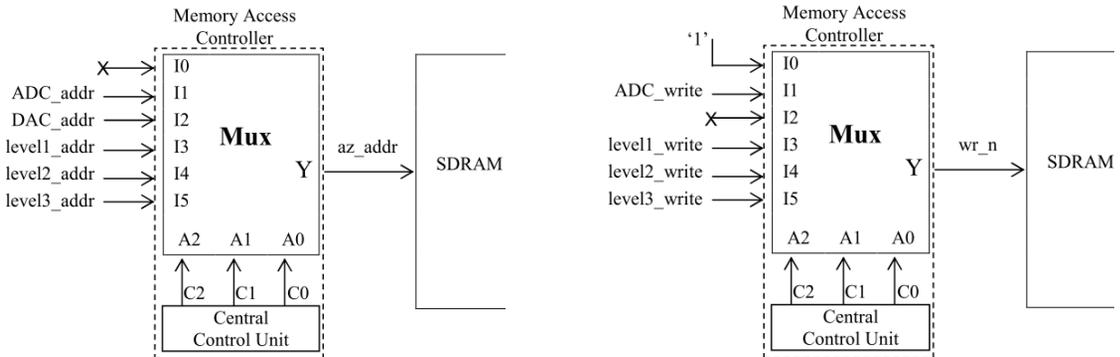
3.3 Memory access controller

Block diagrams are used to depict how to select the control signals for writing into or reading back data from SDRAM, as shown in Figure 5. In this design, the multiplexers select which signals would be sent to the memory. These selections are in according to command signals (C₀, C₁, and C₂)

coming from Central Control Unit (CCU). Besides, a demultiplexer is also used to select which way output data of SDRAM feed to. In this case, Central Control Unit plays a role as an arbitrator to generate all control signals for the multiplexers and demultiplexer running properly. The control signals of the CCU is shown in Table 1.

Table 1: The CCU's control signals

C2	C1	C0	Explanations
0	0	0	Idle state
0	0	1	ADC processing
0	1	0	DAC processing
0	1	1	1 st -level DWT processing
1	0	0	2 nd -level DWT processing
1	0	1	3-rd-level DWT processing



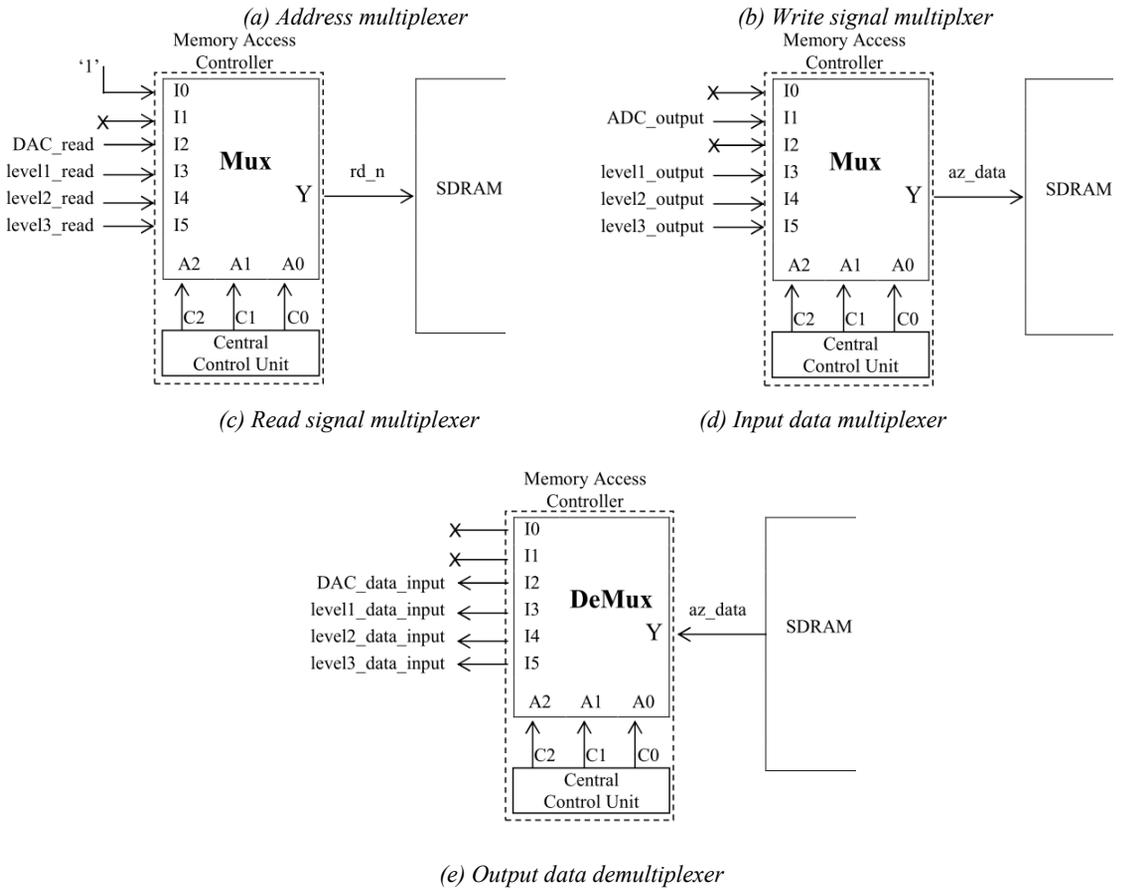


Fig. 5: Block diagrams of Memory Access Controller

3.4 Lifting scheme Db4 discrete wavelet transform

Figure 6 shows the lifting scheme DWT quantization diagram 9. Due to the arithmetic complexity

and slow speed of the floating-point operation in hardware, lifting scheme Db4 DWT coefficients have to be scaled up and then truncated into integers that are shown in the following equations.

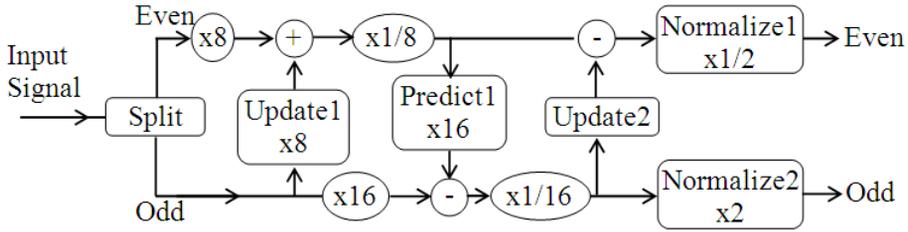


Fig. 6: The lifting scheme Db4 DWT quantization diagram

Update1: (6)

$$Even[n] \approx \frac{8Even[n] + 14Odd[n]}{8}$$

Predict1: (7)

$$Odd[n] \approx \frac{16Odd[n] - 7Even[n] + Even[n-1]}{16}$$

Update2: (8)

$$Even[n] = Even[n] - Odd[n+1]$$

Normalization1: (9)

$$Even[n] \approx 0.5Even[n]$$

Normalization2: (10)

$$Odd[n] \approx 2Odd[n]$$

In this proposed hardware, both the quality of sound and time consuming for processing are considered when selecting the factors. In Equation (6) for Update1 step, the fraction $14/8$ (1.75) is proposed for truncated value of $\sqrt{3}$ (~ 1.732) in

original Equation (1) rather than 2 as (Jing *et al.*, 2008) to improve the result for whole process. Importantly, in the lifting scheme Db4 DWT because the steps are consecutively performed, the error of previous step is seriously amplified at the later one.

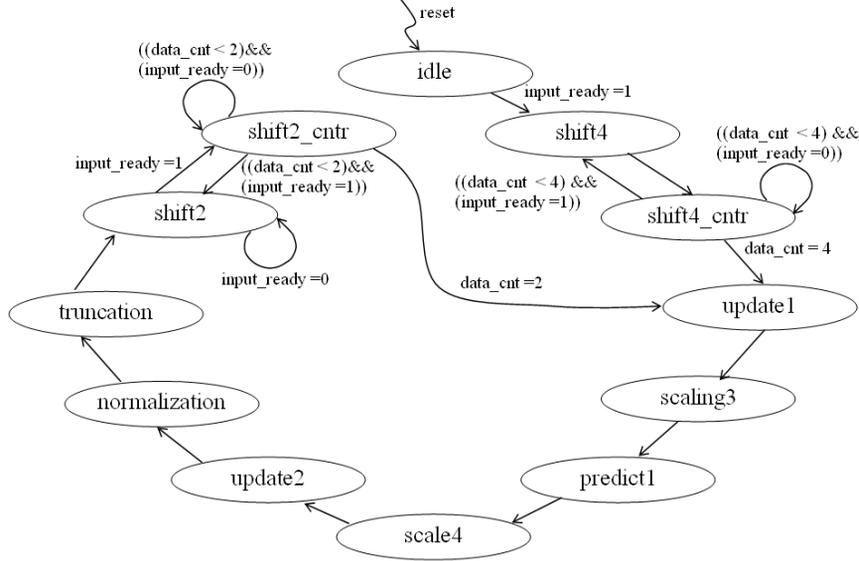
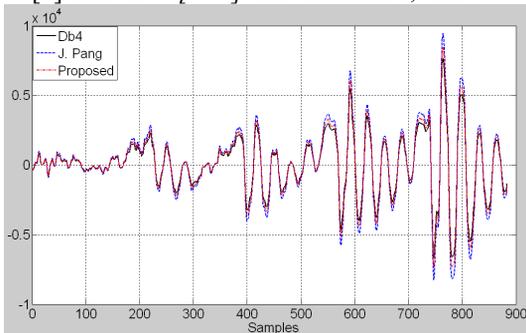


Fig. 7: Finite State Machine (FSM) of forward lifting scheme DWT

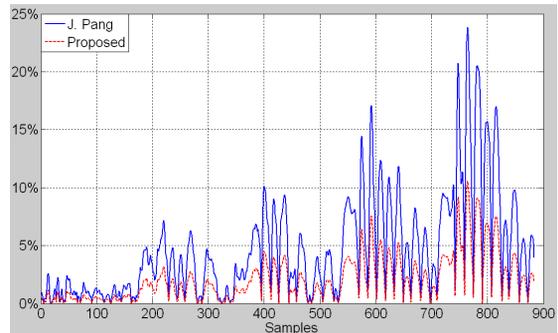
The main part of this design is lifting scheme DWT unit. Figure 7 shows finite state machine (FSM) diagram of it. The *idle* state, where an active *input_ready* signal indicates a 16-bit parallel digital audio sample is available, is utilized to wait for the coming valid digital audio samples. *shift4* and *shift4_ctr* are used to collect four neighboring audio samples in pairs. At *update1* state, both *Even[n]* and *Even[n+1]* are calculated, and then

Odd[n], *Odd[n+1]* are also carried out in *predict1* state. Successively, *update2* state performs to obtain the new *Even[n]* based on *Even[n]* and *Odd[n+1]*. After that, truncate state is employed for truncating the new even component to 16-bit. Finally, two states: *shift2*, *shift2_ctr* are used to collect a pair of audio samples.

4 RESULTS AND CONCLUSION



(a)



(b)

Fig. 8: Experimental results on a typical speech data

The first step of this work is to decide a proper set of DWT factors for hardware implementation, as mentioned in section 3.4. The experimental results were obtained by performing some testing pro-

grams, which were coded in MATLAB. Figure 8(a) illustrates the results on the same axes in case of applying three different set of factors as proposed by Ingrid Daubechies (Ingrid *et al.*, 1998), Jing

Pang (Jing *et al.*, 2008) and this work on a typical speech data respectively. Besides, Figure 8(b) shows absolute deviations in percentage for two comparisons in turn between the results by applying Ingrid Daubechies' factors and Jing Pang's factors, and

then with the our proposed one. Consequently, with a modified factor for *predict* step, the achieved results of applying this work proposed factors is better than the other presented by Jing Pang.

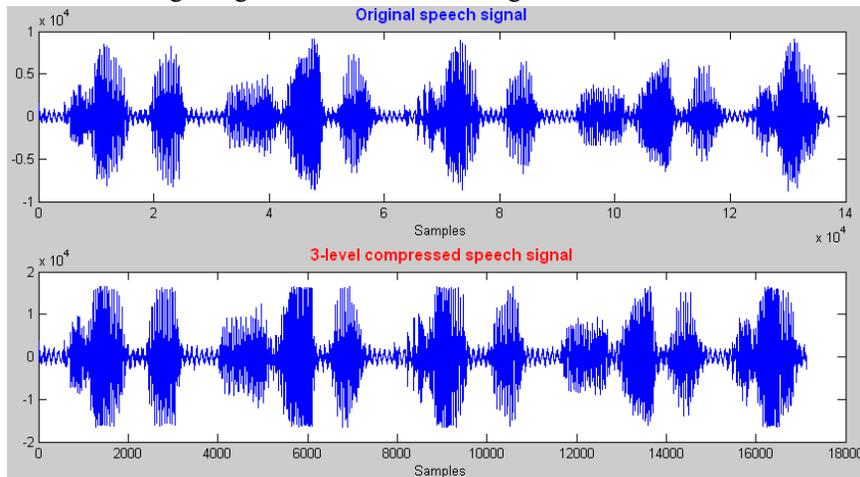


Fig. 9: MATLAB plots for original speech and 3-level compression

Moreover, a program for three-level compression was done on MATLAB. In this test, speech signal was recorded from soudcard on PC. To evaluate the results, the obtain data were plotted for observing easily (as shown in Figure 9) as well as played on loudspeakers. The result is that compressed speech after up-sampling can be heard clearly.

Flow Summary	
Flow Status	Successful - Sun Dec 22 11:14:38 2013
Quartus II Version	10.1 Build 153 11/29/2010 SJ Full Version
Revision Name	DWT
Top-level Entity Name	DWT
Family	Cyclone II
Device	EP2C35F672C6
Timing Models	Final
Total logic elements	1,724 / 33,216 (5 %)
Total combinational functions	1,492 / 33,216 (4 %)
Dedicated logic registers	1,147 / 33,216 (3 %)
Total registers	1213
Total pins	96 / 475 (20 %)
Total virtual pins	0
Total memory bits	0 / 483,840 (0 %)
Embedded Multiplier 9-bit elements	0 / 70 (0 %)
Total PLLs	1 / 4 (25 %)

Fig. 10: FPGA resource usage

Figure 10 shows that the design consumes 1,724 of the total resource available on the EP2C35F672 chip.

The digitized speech signals are compressed immediately after each pair of samples is available. After the 3rd level lifting scheme DWT is performed, the final compressed speech results are written into SDRAM. The whole compression process is done in real time. After that, the compressed data is read back, up sampled by 8, and then played

in the speakers. The compressed speech could be heard clearly without audible background noise. With three levels, compression 87.5% compression is achieved, meaning that 12.5% of original sound components are reduced.

For checking the constraint of compression level in real time processing, speech compression architectures from 2 to 7 levels were implemented. Based on examining the FSM diagram of DWT in Section 3.4, it's easy to figure out that each compression level needs at least 10 processing states. In my design, BCLK was used as clock signal for DWT block. We have also known that ADC of WM8731 takes 64 BCLK clock cycles for each new digital audio even and odd sample pair to be collected (Wolfson Microelectronics Plc., 2004) Consequently, the maximum number of level is 6 because further than this the compression processing cannot be completed before next pair of samples to come. Six-level lifting scheme Db4 DWT real time speech compression is also implemented. The compressed speech is only recognizable, but not very clear. Further than six-level the compressed speech could not be recognized.

Daubechies lifting scheme discrete wavelet transform is not only suitable for compressing non-stationary signals but also carrying out fixed compression ratios. The design of DWT can also be used for several other signals such as seismic sig-

nal, biological signal (Ooninx *et al.*, 2001; Mehmet, 2008).

REFERENCES

- Abdul, M.M.A.N., Abdul, R.R., Azizah, I., Syed, A.R., 2003. Comparing Speech Compression Using Wavelets with Other Speech Compression Schemes. Student Conference on Research and Development (SCOReD) 2003 Proceedings, Putrajaya, Malaysia, pp. 55-58.
- Altera Corporation, 2008. Using the SDRAM Memory on Altera's DE2 Board with Verilog Design, accessed on 01 August 2015. Available from ftp://ftp.altera.com/up/pub/Tutorials/DE2/Computer_Organization/tut_DE2_sdr_am_verilog.pdf
- Hussein, M.M., Nasser, A., Mahmoud, A.O., Alfandi, S.A., 2010. Multimedia Speech compression Techniques. 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT), Chengdu- China, 9-11 July, 2010, pp. 498-502.
- Ingrid, D., Wim, S., 1998. Factoring Wavelet Transform into Lifting Steps. *J. Fourier Anal. Appl.*, 4 (no. 3), pp. 247-269.
- Integrated Circuit Solution Inc., 2000, IS42S8800 / IS42S8800L / IS42S16400 / IS4216400L 2(1) M Words x 8(16) Bits x 4 Banks (64-MBIT) synchronous dynamic RAM, Product Datasheet DR007-0A, pp. 1-68.
- Ivan, K., 2003. The Daubechies D4 Wavelet Transform, accessed on 01 August 2015. Available from http://www.bearcave.com/misl/misl_tech/wavelets/daubechies/index.html
- James, S.W., 1999. A Primer on Wavelets and their Scientific Applications. Chapman and Hall/Crc, pp. 1-58.
- Jing, P., Shitalben, C., Jay, M.B., 2008. Speech Compression FPGA Design By Using Different Discrete Wavelet Transform Schemes. *Advances in Electrical and Electronics Engineering – IAENG Special Edition of the World Congress on Engineering and Computer Science*, pp. 21-29.
- Johnson, I.A., 1996. Discrete Wavelet Transform Techniques in Speech Processing. *IEEE TENCON-Digital Signal Processing Applications*, pp. 514-519.
- Mehmet, R.C., 2008. Comparison of Wavelet and Short Time Fourier Transform Methods in the Analysis of EMG Signals. *Journal of Medical Systems*, Volume 34, Number 1, pp. 91-94.
- Ooninx, P.J., Sleeman, R., Van, E.T., 2001. An Application of the DWT in Seismic Data Analysis. *Wavelets in Signal and Image Analysis Computational Imaging and Vision* Volume 19, pp. 479-500.
- Polikar, R., 1999. The Wavelet Tutorial, accessed on 01 August 2015. Available from <http://users.rowan.edu/~polikar/WAVELETS/WTtutorial.html>
- Rajesh, G., Kumar, A., Ranjeet, K., 2001. Speech Compression using Different Transform Techniques. *International Conference on Computer & Communication Technology (ICCT)*, pp. 146-151.
- Sun, L., Mkwawa, I.H., Jammeh, E., Ifeakor, E., 2013. *Guide to Voice and Video over IP for Fixed and Mobile Networks*. Computer Communications and Networks, Springer-Verlag London, pp. 17-51.
- Wolfson Microelectronics Plc., 2004. WM8731/WM8731L portable Internet audio CODEC with headphone driver and programmable sample rates. Production datasheet, pp. 1-59.
- Wonyong, C., Jongsoo, K., 1997. Speech and Image Compressions by DCT, Wavelet, and Wavelet Packet. *International Conference on Information, Communication and Signal Processing (ICICS '97)*, Singapore, 9-12 September 1997, pp. 1353-1357.