International Journal of Advanced Information Science and Technology (IJAIST) ISSN: 2319:2682
Vol.5, No.1, January 2016 DOI:10.15693/ijaist/2016.v5i1.16-25

Design of Low Complexity Filterbank Method for the State-of-the-art Audio Coding Schemes

Tsung-Han Tsai, Hsin-Chaung Liu and De-Ming Chen
Department of Electrical Engineering,
National Central University, R.O.C

Abstract—**This paper presents an efficient design for filterbanks in variant audio coding standards. Filterbank is generally interpreted as two kinds of approaches, one is modified discrete cosine transform (MDCT) domain and the other is quadrature mirror filterbank (QMF) domain. We propose the fast decomposition methods to solve all of them. For MDCT-based algorithms, they are modified to N-points discrete cosine transform type IV (DCT-IV) with windows cascade operation. We propose the method to decompose the transform matrix operations into conventional N/2-points DCT-IV and DCT-II type. Our method can reduced more than 96% of computation with respect to the original multiplications and additions. For QMF-based algorithms, we not only concern the most popular coding standard, MPEG-1 layer III (MP3), but also address on the newest coding method, the spectral band replication (SBR) for MPEG-4 High Efficiency Advanced Audio Coding (HE-AAC). The computation can be reduced about 90% with respect to the original. Based on these techniques, the computation complexity for most state-of-the-art audio coding schemes can be marvelously reduced. Meanwhile we propose the architecture and VLSI design for these filterbanks by TSMC 90nm library. As a cost-effective design, it consumes about 21K gates with power consumption of 55 mW.**

Index terms –Audio coding, filterbank QME, MDCT, VLSI .

I. INTRODUCTION

Digital audio coding has become more and more popular and represented as essential feature of multimedia consumer electronics in recent years. Various digital audio coding standards have been developed to provide high quality audio compression. They are widely used in many areas from personal computer, network, and mobile phone to portable devices as shown in Table I. Due to the high quality requirement on audio coding, a group of filters is implemented as filterbank, a unique technique applied to audio coding. Filterbank can be interpreted as modified discrete cosine transform (MDCT) domain or quadrature mirror filterbank (QMF) domain. MDCT has been

widely used in state-of-the-art audio coding scheme such as Dolby AC-3, MPEG-2/4 AAC, MP3 [1]-[5]. It is a linear orthogonal lapped transform, based on the idea of time domain aliasing cancellation (TDAC) [6]. MDCT provides critical sampling and overlapping of blocks with good frequency selectivity. It is critically sampled, which means though it is 50% overlapped. A sequence data after MDCT has the same number of coefficients as samples before the transform (after overlap-and-add).

Table I: Audio codec adopted in different applications

Application	Format
DAB	MPEG-1 Layer II, MPEG-2 Layer II
DVB	MPEG-1 Layer I/II, MPEG-2 Layer I/II, HE-AAC
VCD	MPEG-1 Layer II
DVD	MPEG-2 Layer II, AC3
3GPP	Enhanced AAC+
PC, Mobile phone, Consumer audio codec	MP3, WMA, AAC

Referring to QMF method, it performs a filterbank with splitting an input signal into two or more bands. The resulting high-pass and low-pass signals are often decimated by a factor of two, giving a critically sampled two-channel representation of the original signal. QMF is used in MP3, one of the most universal audio standards. Except MP3, QMF has been extended to the newest coding method, spectral band replication (SBR). It is embedded with AAC core [7] to make the MPEG-4HE-AAC to achieve high audio quality at much lower bitrates with increasing complexity [3].

We profile these versatile audio coding systems, including: MP3, MPEG-2/4 AAC, DolbyAC-3 and HE-AAC. Based on this analysis, filterbank indeed takes most of the computation complexity in a generic audio codecs. The complexity analysis is shown as Table II. In traditional codecs such as AC-3, MP3 and AAC, filterbank takes more than half of the computation complexity, even up to 83%. In advanced codec, HE-AAC, the analysis and synthesis QMF are summarized as to 80% of overall complexity. Consequently fast algorithm on filterbank is very crucial to reduce the complexity of overall system.

The fast algorithms on MDCT/IMDCT have been proposed for years. Generally they are realized with the following techniques:

(a) the transform kernel of MDCT/IMDCT factorized into the formula of FFT; (b) the trigonometric equivalence to convert the coefficients of transform kernel into twiddle factor form recursively; (c) the matrix decomposition to reduce size from N to N/2 and accomplish the formula by DCT/IDCT-II kernel, and

(d) the trigonometric equivalence mapping the kernel to DCT-II and achieving the computation by fast DCT algorithm. Fast MDCT algorithm is based on a cascade of window operators

Table II: Complexity analysis of state-of- the art decoders

	Function	Complexity
MP3	Huffman	1.90%
	IQ	38.70%
	Stereo	3.40%
	Antialiasing filter	0.60%
	DWIMDCT filterbank	21.50%
	Polyphase filterbank	33.90%
AAC	Huffman	19.60%
	IQ	1.60%
	Rescale	1.90%
	Stereo	2.70%
	TNS	0.60%
	Filterbank	73.60%
AC-3	Decode Envelop	1.30%
	Bit Allocation	1%
	Mantissa	2%
	Filterbank	1%
HE-AAC	Core AAC	16%
	HE generator	1%
	Envelope adjuster	1%
	other	2%
	AQMF	17%
	SQMF	63%
HE-AAC Downsampled version	Core AAC	12%
	HE generator	2%
	Envelope adjuster	2%
	other	3%
	AQMF	39%
	Downsampled SQMF	42%

and a discrete cosine transform type IV (DCT-IV) [6]. The efficient algorithm of DCT-IV can be classified to two categories which are direct computation and indirection computation. The direct computation reduces computational complexity by matrix factorization and recursive decomposition. Indirect computation takes the advantage of exiting fast algorithm such as FFT to compute DCT [8]-[9]. However, additional operations are often required for mapping the DCT sequence to other transformation sequence.

With respect to QMF-based methods, some of them have been derived with the fast algorithm in [10]. In [11] presented the low power SBR with aliasing minimizing tools to reduce the

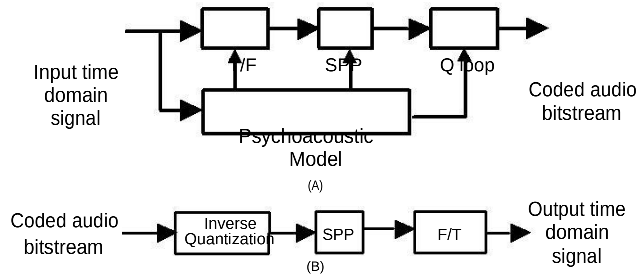

Fig. 1. Block diagram of general audio encoder/decoder. (a). Audio encoding flow, (b). audio decoding flow.

complexity of the decoder by processing real-valued data instead of complex-valued data. However, the complexity is still high compared with the conventional plain AAC decoder if SBR is not optimized. In [12], they presented a fast algorithm and common structure design to deal with mirror filterbanks (AQMF, SQMF) on SBR. Since they used a simple approach with recursive computation method, only few reduction of computation complexity is obtained.

In this paper, we provide an algorithm level solution to improve the performance for state-of-the-art audio coding applications. Our goal is to derive the fast algorithm to reduce the complexity of the computation-intensive matrix operations, and interpret it as a universal method for all the filterbanks in audio coding standards. The proposed method comprises only conventional discrete cosine transform of type II and III (DCT-II and DCT-III) [13] with simple permutations. By deriving to a unified DCT computations, many conventional fast DCT algorithms can be effectively used. For MDCT domain method, it is modified to DCT-IV with windows cascade operation. DCT-IV will be further modified to DCT-II and discrete sine transform type II (DST-II) [14]. For QMF domain method, the proposed method is based on the direct decomposition structure to modify DCT-II and DCT-III [15] structure to save arithmetic operator. Furthermore we provide the architecture based on the proposed algorithm, and manipulate it as an intellectual property (IP). The paper is organized as follows. In Section II, algorithm on MDCT and QMF is briefly introduced. In Section III, the decomposition method for reducing the computational complexity is proposed. Section IV, the performance evaluation and the VLSI design is derived. Finally we make a conclusion on this topic.

II. OVERVIEW OF ALGORITHMS IN MDCT AND QMF

Fig.1 illustrates the flowchart of encoder and decoder in audio compression. The decoder flow includes inverse quantization (IQ), spectrum processor (SPP), and frequency to time domain conversion (F/T conversion). The encoder flow is similar to decoder, excluding psychoacoustic model (PAM) module, which models the human auditory system and keeps audio quality in encoder scheme [1].

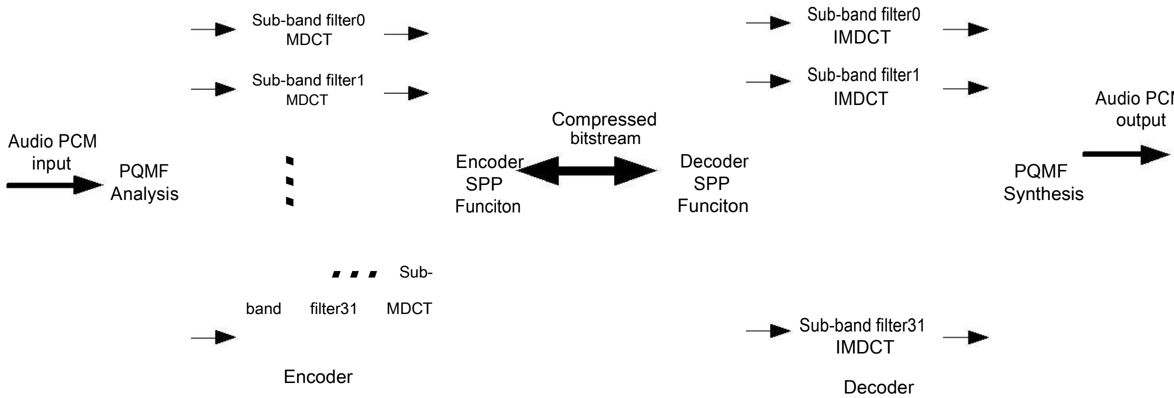

Fig. 2. Block diagram of the MPEG-1 Layer 3Codec with specifying PQMF and MDCT.

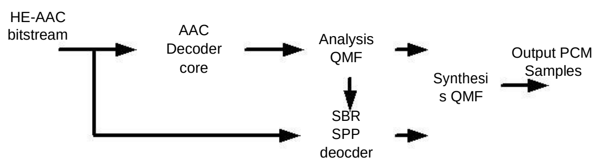

Fig. 3. Block diagram of the HE AAC decoder with the SBR decoder.

As shown in Fig. 1, the frequency/time domain conversion requires MDCT, especially in AAC and AC-3 encoder/decoder flow. Generally audio coding technology differs from image or video processes. It applies MDCT instead of DCT to achieve spectrum conversion. MDCT have some property mostly proper to apply in audio coding. It can reduce or cancel the noise and alias signal, and the spectrum response after converting is similar to human acoustic characteristic. MDCT converts data from 2N-point to N-point with this property and combines windows overlap operation to cancel time domain alias without inducing extra complexity. AAC and AC-3 apply the same operator MDCT to convert signal spectrum, where AAC applies 2048/256 point operation and AC-3 applies 512/256 point respectively.

MP3 is developed before pure MDCT-based AAC coding. It exploits transform domain coding technique and subband coding scheme as a hybrid subband coding. Except MDCT, polyphase QMF (PQMF) technique is also included in MP3. As the most popular audio standard, we depict the block diagram for MP3 coding flow in Fig. 2. Recently, QMF-based method has been used in SBR coding to construct a higher efficient audio coding. Fig. 3 illustrates the block diagram of the MPEG HE-AAC decoder. It displays how the SBR decoder and the core AAC decoder are interconnected. The data of SBR are processed in the QMF domain which is the output of the prime AAC decoder. Firstly, it analyzed with a 32-channel analyzing QMF (AQMF) filterbank. Then SBR decoder processed QMF-domain signal and reconstructed transient audio signal. Finally, time-domain audio output is synthesized from lower band and higher band with a 64-channel synthesis QMF (SQMF). In order to circumvent the situations where the output signals have a sampling rate twice than that of the input, the downsampled 32-channel SQMF is employed instead of the 64-channel SQMF.

.

2.1 MDCT Methods

The MDCT formula is

$$X_t(k) = \sqrt{\frac{2}{N}} \sum_{m=0}^{2N-1} w(k)x_t(m)\cos\left(\frac{\pi}{4N}(2m+1)(2k+N+1)\right)$$

for $m = 0,1,2.......N-1$ (1)

And the IMDCT formula is

$$x_t(k) = w(k)\sqrt{\frac{2}{N}} \sum_{m=0}^{N-1} X_t(m)\cos\left(\frac{\pi}{4N}(2m+1)(2k+N+1)\right)$$

for $k = 0,1.....2N-1$ (2)

MDCT will introduce the time domain aliasing. The aliasing error can cancel by overlap and add method of output data between two succeeding block t and t+1 of inverse MDCT. The overlapping part is:

$$x'_t(k) = y_t(N+k) + y_{y+1}(k)$$

for $k = 0,1.....N-1$

To ensure this time domain aliasing cancellation, the windows of two succeeding blocks have to fulfill certain condition in their overlapping part. A sufficient condition for time domain aliasing cancellation is

$w(k)^2 + w(N+k)^2 = 1$ and $w(k) = w(2N-1-k)$

for $k = 0,1.....N-1$

2.2 QMF Methods

QMF is applied with various forms. The matrix operations (scale factors are neglected) in the QMF are defined as follows. PQMF (MP3-Analysis):

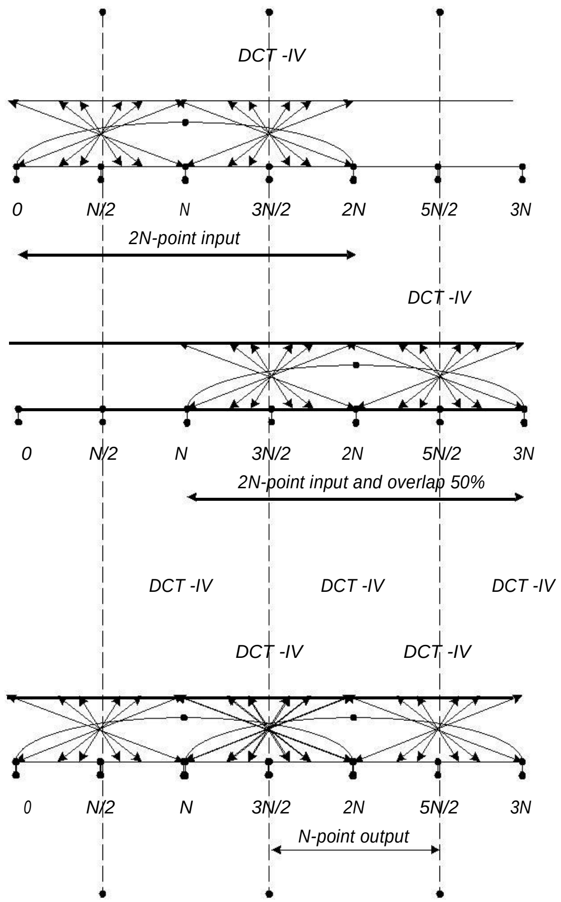

Fig. 4. The symmetry property on transform and window operation for MDCT.

$$X(k) = \sum_{n=0}^{63} u(n) \cos\left[\frac{\pi}{64}(n-16)(2k+1)\right]$$

for k=0,1,…,31; n=0,1,…,63 (3)

PQMF (MP3-Synthesis):

$$v(n) = \sum_{k=0}^{31} u(n) \cos\left[\frac{\pi}{64}(n+16)(2k+1)\right]$$

for k=0,1,…,31; n=0,1,…,63 (4)

QMF (HE-AAC-Analysis):

$$X(k) = \sum_{n=0}^{63} u(n) \cos\left[\frac{\pi}{64}(n-48)(2k+1)\right]$$

for k=0,1,…,31; n=0,1,…,63 (5)

QMF (HE-AAC-Synthesis):

$$v(n) = \sum_{k=0}^{63} X(k) \cos\left[\frac{\pi}{128}(n-32)(2k+1)\right]$$

for k=0,1,…,63; n=0,1,…,127 (6)

QMF (HE-AAC-synthesis downsampled):

$$v(n) = \sum_{k=0}^{31} X(k) \cos\left[\frac{\pi}{64}(n-16)(2k+1)\right]$$

for k=0,1,…,31 ;n=0,1,…,63 (7)

III. PROPOSED METHODS

DCT is one of the most widely used transforms in digital signal processing application such as data compression for image, video, audio, and speech signal [15], [16]. According to [13], the family of DCT consists of 8 versions of DCT and corresponding 8 versions of DST. Each transform is identified as even or odd and of type I, II, III, and IV. Here, only DCT-II and DCT-III are considered because they are more popular and easily employed.

3.1 DCT-II, DCT-III and DST-II

The relationship of the DCT and DST is quite close. The formula of DCT-II DCT-III, and DST-II are shown as

(DCT-II): $$Xc(m) = \sum_{k=0}^{N-1} u(k) \cos\left(\frac{\pi}{2N}(m)(2k+1)\right)$$

for $m = 0,1.....N-1$

(DCT-III): $$Xc(m) = \sum_{k=0}^{N-1} u(k) \cos\left(\frac{\pi}{2N}(k)(2m+1)\right)$$

for $m = 0,1.....N-1$

(DST-II): $$Xs(m) = \sum_{k=0}^{N-1} u(k) \sin\left(\frac{\pi}{2N}(m)(2k+1)\right)$$

for $m = 0,1.....N-1$

Briefly, DST-II is very similar with DCT-II. After matrix permutation DST-II is the same as DCT-II [17]. A translation from DST-II to DCT-II is a matrix operation as:

$$[S_N] = [\overline{I_N}][C_N][D_N]$$

S_N is DST-II and C_N is DCT-II. $[\overline{I_N}]$ is the opposite diagonal identity matrix and $[D_N]$ is special matrix defined as follow :

$$[D_N] = \begin{bmatrix} 1 & & & 0 \\ & -1 & & \\ & & 1 & \\ 0 & & & \end{bmatrix}$$

3.2 MDCT using DCT-IV and DCT-II

The symmetry property on transform and window operation is existed for MDCT, as illustrated in Fig. 4. It shows that 2N-point MDCT is shaped by window operation. With 50% overlap on MDCT, it can be derived to N-point DCT-IV. From (1), it can be modified into overlapping windows operation with DCT-IV as

International Journal of Advanced Information Science and Technology (IJAIST) ISSN: 2319:2682
Vol.5, No.1, January 2016 DOI:10.15693/ijaist/2016.v5i1.16-25

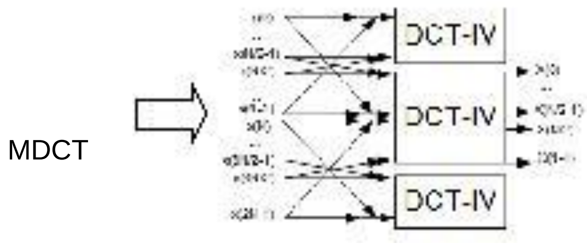

Fig. 5. Decomposition of MDCT to DCT-IV.

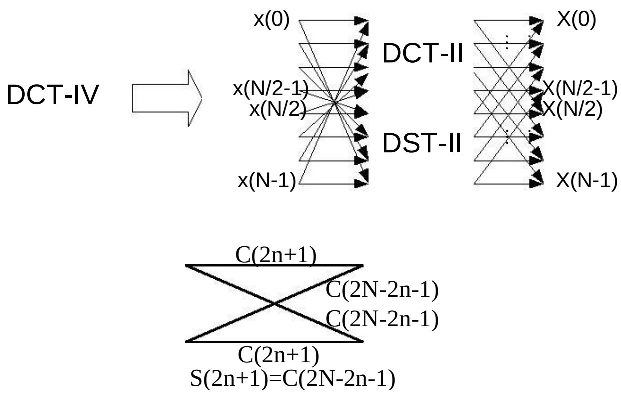

Fig. 6. Decomposition of DCT-IV to DCT-II and DST-II.

$$Xc(m)=\sum_{k=0}^{N-1} u(k)\cos\left(\frac{\pi}{4N}(2m+1)(2k+1)\right)$$

for $m=0,1.....N-1$

where

$$u(k+\frac{N}{2})=w(N-1-k)x_t(N-1-k)-w(k)x_t(k)$$

$$u(\frac{N}{2}-1-k)=w(k)x_t(N-1-k)+w(N-1-k)x_t(k)$$

for $k=0,1.....\frac{N}{2}-1$

MDCT can be modified into DCT-IV and then it can be simplified as Fig. 5. DCT-IV can be built as an orthogonal matrix. The matrix is decomposed into N(N-1)/2 givens rotations [20], but the decomposition method is not unique. Thus we try to find an efficient decomposition method for computing MDCT and DCT-IV.

Since DCT-II is easily implemented than DCT-IV, we try to find an efficient method to modify DCT-IV to DCT-II. The equation of DCT-IV is shown as

$$Xc(m)=\sum_{k=0}^{N-1}u(k)\cos\left(\frac{\pi}{4N}(2m+1)(2k+1)\right)$$

for $m=0,1.....N-1$ (8)

By the trigonometric identify

$$\begin{cases}\cos(A+B)=\cos A\times\cos B-\sin A\times\sin B\\ \sin(A+B)=\sin A\times\cos B+\cos A\times\sin B\end{cases}$$

where

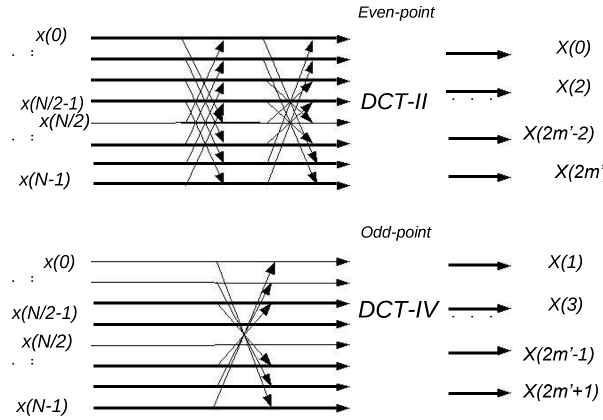

Fig. 7. Merge DCT-II and DST-II, and map to N/2-DCT-II, and N/2-DCT-IV.

$$\begin{cases}A=(2k+1)\pi/4N\\ B=(2k+1)m\pi/2N\end{cases}$$

As referred in [19], (8) is replaced by DCT-II and DST-II and shown in Fig. 6. Then it is reformed as

$$Xc(m)=\sum_{k=0}^{k=N/2-1}(u(k)\cos A+u(N-1-k)\sin A)\cos(B)+(-1)^{k+1}\sum_{k=0}^{k=N/2-1}(u(N-1-k)\cos A-u(k)\sin A)\sin(B)\tag{9}$$

Consequently the equation becomes a N/2-point DCT-II and N/2-point DST-II. Using the representation of DCT-II and DST-II used in [17], we can obtain

$$Xc(m)=\sum_{k=0}^{k=N/2-1}(u(k)\cos A+u(N-1-k)\sin A)\cos(B)+\sum_{k=0}^{k=N/2-1}(u(N-1-k)\cos A-u(k)\sin A)\cos(\frac{(2k+1)(N-m)\pi}{2N})\tag{10}$$

After using the permutation of method in [18], the equation can be rewritten as

$$Xc(m)=\sum_{k=0}^{k=N/2-1}(u(k)\cos A+u(N-1-k)\sin A)\cos(B)-\sum_{k=0}^{k=N/2-1}(u(N-1-k)\cos A-u(k)\sin A)\cos(B)\tag{11}$$

Then (11) can be rewritten as (12).

$$Xc(m)=\sum_{k=0}^{k=N/2-1}((u(k)-u(N-1-k))\cos A\cdot\cos(B)+\sum_{k=0}^{k=N/2-1}(u(N-1-k)+u(k))\sin A\cdot\cos(B)\tag{12}$$

After separating the odd and even term as in [20], it becomes as:

$$Xc(2m')=\sum_{k=0}^{k=N/2-1}h(k)\cdot\cos(\frac{(2k+1)\cdot 2m'\pi}{2N})$$

International Journal of Advanced Information Science and Technology (IJAIST) ISSN: 2319:2682
Vol.5, No.1, January 2016 DOI:10.15693/ijaist/2016.v5i1.16-25

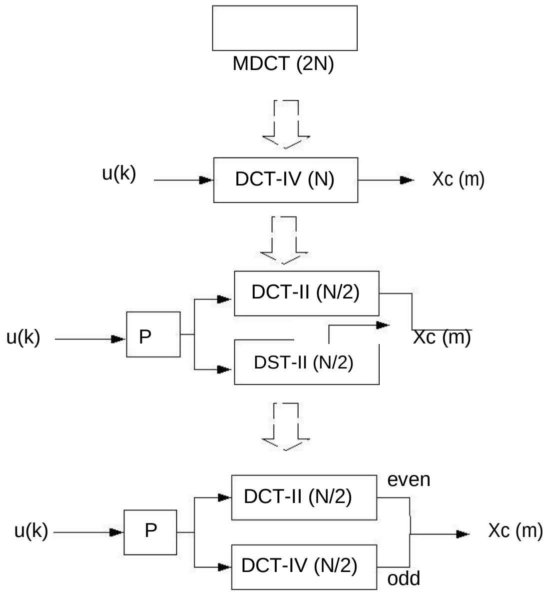

Fig. 8. Decomposition of N-point DCT-IV.

$$Xc(2m'+1)=\sum_{k=0}^{N/2-1} g(k)\cdot\cos\left(\frac{(2k+1)(2m'+1)\pi}{2N}\right)$$

For $m'=0,1.....N/2-1$ (13)

where *g(k) =(u(k)-u(N-1-k))cosA* and *h(k)=(u(k)+u(N-1-k))sinA*

(13) represents a N/2-point DCT-II for even part and N/2-point DCT-IV for odd part, as shown in Fig. 7. Based on (13), this means the transform can be expressed as

$$X_N^{IV}(m)=X_{N/2}^{IV}(2m'+1)+X_{N/2}^{II}(2m')$$

As a result, 2N-point MDCT transform can be decomposed into two N/2-point terms, where one is by DCT-II and the other is by DCT-IV. Then DCT-IV can be recursively decomposed into half-point one. The decomposition flow is shown as Fig. 8.

3.3 QMF using DCT-II and DCT-III

Implementation on DCT-II or DCT-III is easier than the original QMF form. Thus we try to interpret an efficient method to modify QMF into DCT-II or DCT-III.

A. PQMF (Analysis)

First, let (14) be defined as

$$X(k)=\sum_{n=0}^{63} u'(n)\cos\left[\frac{\pi}{64}n(2k+1)\right]$$

for k=0,1,…,31 ; n=0,1,…,63 (14)

Then the relationship between u'(n) in (14) and u(n) in (3) is derived.

Let j=n-16, then (3) is rewritten as

$$X(k)=\sum_{j=-16}^{47} u(j+16)\cos\left[\frac{\pi}{64}j(2k+1)\right]$$

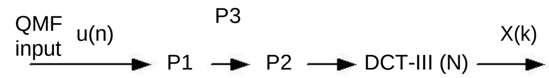

Fig. 9. General decomposed matrix flow of PQMF (analysis).

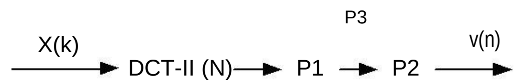

Fig. 10. General decomposed matrix flow of PQMF (synthesis).

$$=\sum_{j=-16}^{-1} u(j+16)\cos\left[\frac{\pi}{64}j(2k+1)\right]+\sum_{j=0}^{47} u(j+16)\cos\left[\frac{\pi}{64}j(2k+1)\right]$$

(15)

Substituting m=j+64 into the above, it yields

$$X(k)=\sum_{m=48}^{63} u(m-48)\cos\left[\frac{\pi}{64}(m-64)(2k+1)\right]+\sum_{j=0}^{47} u(j+16)\cos\left[\frac{\pi}{64}j(2k+1)\right]$$
$$=\sum_{m=48}^{63}(-u(m-48))\cos\left[\frac{\pi}{64}m(2k+1)\right]+\sum_{j=0}^{47} u(j+16)\cos\left[\frac{\pi}{64}j(2k+1)\right]$$

(16)

Combining (14) and (16), the relationship can be described as (17), which stands for P1 in Fig. 9.

$$u'(n)=\begin{cases} u(n+16) & n=0,1,...,47 \\ -u(n-48) & \text{for } n=48,49,...,63 \end{cases}$$ (17)

Second, let (18) be defined as 32-point DCT-III

$$X(k)=\sum_{n=0}^{31} u''(n)\cos\left[\frac{\pi}{64}n(2k+1)\right]$$

for k=0,1,…,31 ;n=0,1,…,31 (18)

Then the relationship between u'(n) in (17) and u''(n) in (18) can be derived as:

$$=u'(0)+\sum_{n=1}^{31} u'(n)\cos\left[\frac{\pi}{64}n(2k+1)\right]+\sum_{n=1}^{31} u'(64-n)\cos\left[\frac{\pi}{64}(64-n)(2k+1)\right]$$
$$=u'(0)+\sum_{n=1}^{31}(u'(n)-u'(64-n))\cos\left[\frac{\pi}{64}n(2k+1)\right]$$

(19)

Combining (18) and (19), the relationship can be described as fallowing, which stands for P2 in Fig. 9.

$$u''(n)=\begin{cases} u'(0) & n=0 \\ u'(n)-u'(64-n) & \text{for } n=1,2,...,31 \end{cases}$$ (20)

Combining (17) and (20), the relationship can be described as (21), which stands for P3 in Fig. 9.

Table III: Decompositions of different filterbanks

Filterbank	MDCT (AAC)	MDCT (AC-3)	MDCT (MP3)	SQMF
Decomposition type	DCT-II(512)+ DCT-IV(512)	DCT-II(128) + DCT-IV(128)	DCT-II(9)+ DCT-IV(9)	DCT-II(64)
Filterbank	PQMF(Analysis)	PQMF (Synthesis)	AQMF	SQMF (Downsampled)
Decomposition type	DCT-III(32)	DCT-II(32)	DCT-III(32)	DCT-II(32)

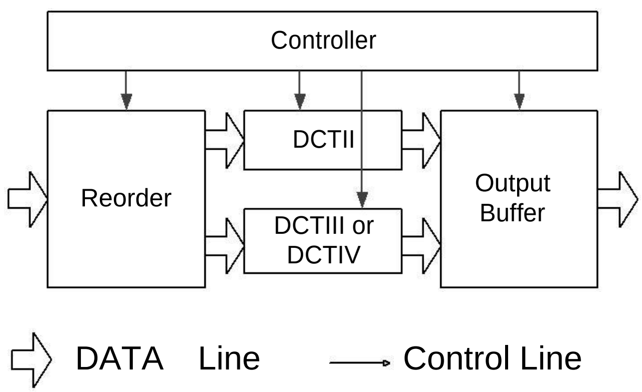

Fig. 11. Block diagram of the proposed method for various filterbanks.

$$u''(n)=\begin{cases} u(16) & n=0 \\ u(n+16)-u(80-n) & n=1,2,...,16 \\ u(n+16)+u(16-n) & \text{for } n=17,...,31 \end{cases} \quad (21)$$

B. PQMF (Synthesis)

First, let (22) be defined as

$$v'(n)=\sum_{k=0}^{31} X(k)\cos\left[\frac{\pi}{64}(n)(2k+1)\right]$$

for k=0,1,…,31 ;n=0,1,…,63 (22)

Then, the relationship between v'(n) in (22) and v(n) in (4) is derived. Therefore, (23) stands for P1in Fig. 10 as following:

$$v(n)=\begin{cases} v'(n+16) & n=0,1,...,15 \\ -v'(n+16) & \text{for } n=16,17,...,63 \end{cases} \quad (23)$$

Second, let (24) be defined as 32-point DCT-II

$$v''(n)=\sum_{k=0}^{31} X(k)\cos\left[\frac{\pi}{64}(n)(2k+1)\right]$$

for k=0,1,…,31 ; n=0,1,…,31 (24)

Then the relationship between v''(n) in (24) and v'(n) in (22) is derived.

For n=0, 1,…, 31, v''(n)=v'(n). For n=32, v''(n)=0. And for n=32,33….,63

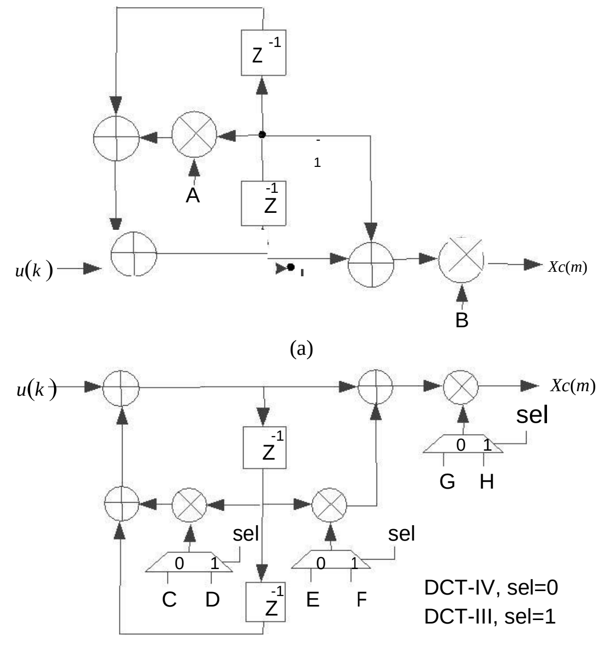

Fig. 12 The proposed architecture for (a) DCT-II, (b) DCT-III and DCT-IV.

$$v'(n)=\sum_{k=0}^{31} X(k)\cos\left[\frac{\pi}{64}(n)(2k+1)\right]=-\sum_{k=0}^{31} X(k)\cos\left[\frac{\pi}{64}(64-n)(2k+1)\right]$$

Therefore, (25) stands for P2 in Fig. 10.

$$v'(n)=\begin{cases} v''(n) & n=0,1,...,31 \\ 0 & n=32 \\ -v''(64-n) & \text{for } n=33,34,...,63 \end{cases} \quad (25)$$

Combining (23) and (24), it yields the direct relationship as:

$$v(n)=\begin{cases} v''(n+16) & n=0,1,...,15 \\ 0 & n=16 \\ -v''(64-n) & n=17,18...,47 \\ -v''(n-64) & \text{for } n=48,49,..,63 \end{cases} \quad (26)$$

As a result, (26) stands for P3 in Fig. 10. PQMF (Analysis/Synthesis) is decomposed into the input permutation and a 32-point DCT-III/DCT-II. In [15], AQMF is decomposed into an input permutation and a 32-point DCT-III. The 64-channel SQMF is decomposed into a 64-point DCT-II and an output permutation, and the downsampled 32-channel SQMF is decomposed into a 32-point DCT-II and an output permutation.

IV. PERFORMANCE AND COMPLEXITY

4.1 Performance and Complexity Evaluation

The decomposed components for each filterbank are listed in Table III. Referring to the performance, it can be estimated by computational arithmetic operations, including the number of multiplications and additions operations. It is known that the

International Journal of Advanced Information Science and Technology (IJAIST) ISSN: 2319:2682
Vol.5, No.1, January 2016 DOI:10.15693/ijaist/2016.v5i1.16-25

Table IV: Evaluation on arithmetic operation

Filterbank	Operator	Original	Proposed			Improved
			Fast DCT	Permutation	Total	(Proposed/Original)
MDCT - AAC (N=2048)	Mult	8388608	5120	0	5120	0.50%
	Add	8384512	13313	2048	15361	0.80%
MDCT - AC-3 (N=512)	Mult	524288	1024	0	1024	2%
	Add	523264	2561	512	3073	3%
MDCT-MP3 (N=36)	Mult	2592	45	0	45	3%
	Add	2556	42	36	78	4%
PQMF –Analysis (N=64)	Mult	2048	160	0	160	3.90%
	Add	2016	160	31	191	11.90%
PQMF –Synthesis (N=64)	Mult	2048	80	0	80	4%
	Add	1984	209	15	224	11%
AQMF [15] (N=64)	Mult	2048	160	0	160	3.90%
	Add	2016	160	31	191	11.90%
SQMF [15] (N=64)	Mult	8192	192	0	192	2.30%
	Add	8064	513	31	544	6.70%
SQMF-Downsampled [15] (N=64)	Mult	2048	80	0	80	3.90%
	Add	1984	209	15	224	11.30%

fast algorithm for N-point DCT-IV requires 3N multiplications and 2N additions, and N-point DCT-II requires 2N multiplications and 2N+4 additions based on the second order shift property [21]. The proposed structure can be decomposed into two N/2-point DCT-II and DCT-IV. Then N/2-point DCT-IV can be recursively decomposed to half-point one. Our analysis for the total arithmetic operators consists of DCT-IV and permutation matrix. The result is shown in Table IV. As the expectation, the overhead on permutation is just only few additions, and do not induce any multiplications. Investigation on these AAC-based algorithms, the computational complexity can be reduced more than 96% with respect to the original one on multiplications and additions.

Table IV also shows the result for QMF. We successfully derive QMF into conventional DCT-II or DCT-III. There are many fast algorithms for N-point DCT-II and DCT-III where N = 2M and M > 0 [22] [23]. The results show that the permutation requires only zero multiplication and little additions. For QMF-based algorithms, the computational complexity can be reduced about 90% with respect to the original multiplications and additions.

4.2 Architecture and VLSI Design

Based on the decomposed components for each filterbank in Table III, three important modules are used including DCT-II, DCT-III, and DCT-IV. DCT-based architectures are established in some previous literature [24]-[26]. We design a unified architecture as illustrated in Fig. 11. The reorder module performs the permutation used in MDCT and QMF method. DCT-II is commonly applied for all filterbanks. DCT-III and IV is selected and used for QMF and MDCT respectively. Furthermore we propose a fixed-coefficient recursive structure for DCT-II shown in Fig. 12(a). It is designed as a recursive structure to save the hardware cost. Only two multipliers and three adders are needed. In Fig. 12(b) we present the architecture for DCT-III and DCT-IV. It is composed with three multipliers and three adders, and with some configuration on coefficients to implement DCT-III and DCT-IV. The parameters are list as in Table V.

Table VI lists the hardware cost and the comparison with other designs. In [27], [28], [29] and [30], those designs only perform MDCT and IMDCT. The proposed can achieve various kinds of filterbanks to cover most of the audio coding codecs. Table VII shows the chip summary of the proposed method. It is synthesis by TSMC 90nm process. As a cost-effective design, it consumes about 21K gates with power consumption of 55 mW.

Table V: The parameters used in proposed architecture

Parameter	Sel	Formula
A	-	$2\cos(\frac{4k\pi}{N})$
B	-	$((-1)^{\frac{k}{2}+1})\cdot 2\cdot\cos(\frac{4k\pi}{N})$
C	0	$2\cdot\cos(\frac{(K+\frac{1}{2})\pi}{N})$
C	1	$2\cdot\cos(\frac{\pi}{2N})$
D	0	1
D	1	-1
E	0	$(-1)^k\cdot(\frac{2}{N})^{\frac{1}{2}}\cdot\sin((k+\frac{1}{2})(\frac{\pi}{2N}))$
E	1	1

Table VI: Hardware cost and comparison results

	Proposed	[27]	[28]	[29]	[30]
Latches	4	3	2	5	3
Adders	6	3	3	9	5
Multipliers	5	2	3	6	3
Computation cycles for N-point input data	N/4	N	N	N/4+1	N/4
Types of filterbank	MDCT SQMF PQMF(Analysis) PQMF(Synthesis) AQMF SQMF (Downsampled)	MDCT IMDCT	MDCT IMDCT	MDCT IMDCT	MDCT IMDCT

Table VII: The synthesis results

	Proposed architecture
Process	TSMC90 *nm*
Gate count	21.27 K
Power	55.25 mW
ROM	10.24 KB
RAM	9.21 KB

V. CONCLUSION

This paper aims to reduce the computation-intensity functions such as MDCT, PQMF, AQMF, and SQMF in different audio standards. We have accomplished the original forms on MDCT and QMF-based filterbank to a general form. For MDCT-based method, we modify it to DCT-IV with windows cascade, and then derive to half-point DCT-II and DCT-IV. For QMF-based method, our main concept is to transform the computation-intensive matrix operations in QMF into conventional DCT, since implementation on DCT-II or DCT-III is easy and highly referred with fast algorithms. Obviously, we have accomplished the marvelous reduction on computation complexity. More than 96% computation complexity for MDCT-based methods and more than 90% for QMF-based methods are reduced. Additional achievements are also revealed. First, we derive these various filterbanks into general DCT forms to achieve high compatibility. Second, in most recent perceptual audio coders, higher compression ratio has been made possible by increasing frequency resolution of the sub-band signals. Our method is independent of it especially it can be recursively decomposed.

Furthermore we proposed a unified architecture which can be applied to many kinds of filterbanks with low complexity and fast computation. The proposed architecture is feasible for very large scale integration architecture and integrated into modern audio codec systems.

REFERENCES

[1]. S. W. Huang, T. H. Tsai, and L. G. Chen, “A Low Complexity Design of Psycho-Acoustic Model for MPEG-2/4 Advanced Audio Coding” IEEE Trans. Consumer Electronics vol. 50, pp.1209-1217, Nov. 2004.

[2]. “Digital audio compression (ac-3) standard,” Document of Advance Television Systems Committee (ATSC), Audio Specialist Group T3/T7,Dec.20.1995.

[3]. “Information technology-generic coding of moving pictures and associates audio-part3:audio,” Document of Advance Television Systems Committee (ATSC), Audio Specialist Group T3/T7,Dec 1995.

[4]. “Information technology-coding of moving pictures and associated audio for digital storage media at up to about 1.5mbit/s -part 3:audio” ISO/IEC/JTC1/SC29/WG11 MPEG, IS11172-3(MPEG-1), 1992.

[5]. V. Britanak, “New fast algorithms for the low delay MDCT computation in the MPEG-4 AAC enhanced low delay audio coding standard” European Association for Signal Processing (EURASIP), Journal on Signal Processing, Dec 2014.

[6]. R. Geiger, T. Sporer, J. Koller, and K. Brandenburg, “Audio Coding based on Integer Transform,” in AES 111th Convention, New York, NY, USA p.Preprint 5471 Sep. 2001.

[7]. R. .K. Chivukula, “Fast Algorithms for Low-Delay SBR Filterbanks in MPEG-4 AAC-ELD” IEEE Transactions on Audio, Speech, and Language Processing, March 2012.

[8]. T. H. Tsai and C. N. Liu, “A Configurable Common Filterbank Processor for Multi-Standard Audio Decoder,” IEICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences, vol.E90-A pp.1913-1923, Sept. 2007.

[9]. P. Duhamel, Y. Mahieux, and J. P. Petit, “A fast algorithm for the implementation of filter banks based on `time domain

aliasing cancellation',” in Proc. IEEE ICASSP, Toronto, vol. 3 , pp. 2209-2212, May 1991.

[10]. S. W. Huang, T. H. Tsai, and L. G. Chen, “Fast decomposition of filterbanks for the state-of-the-art audio coding,” IEEE Signal Process. Lett., vol. 12, no. 10, pp. 693-696, 2005.

[11]. O. Shimada, et al. “A low power SBR algorithm for the MPEG-4 audio standard and its DSP implementation,” Convention Paper 6048 in AES 116th Convention, May 2004.

[12]. A. K. Li, S. F. Lei, W. K. Tsai, S. C. Lai, “Fast algorithm and common structure design of recursive analysis and synthesis quadrature mirror filterbanks for digital radio mondiale,” IEEE Circuits and Systems (ISCAS), 2014 IEEE International Symposium on, 2014, Pages: 1199 – 1202.

[13]. K. R. Rao and P. C. Yip, The transform and data compression handbook, Boca Raton, CRC Press LLC, 2001.

[14]. Z. Zhijin, and Q. Huisheng, “Recursive algorithm for discrete cosine transform,” in Proc. IEEE ICSP, Beijing vol. 1, pp. 115-118, Oct. 1996.

[15]. N. Ahme, T. Natarajan, and K. R. Rao, “Discrete cosine transform,” IEEE Trans. Communication. vol.com-23, pp.90-93, Jan. 1974.

[16]. Z. D. Wang, “Fast algorithm for the discrete sine transform implemented by the discrete cosine transform,” IEEE Trans Acoustic. Speech, Signal Processing. vol.ASSP-30, pp.814-815, Oct. 1982

[17]. P. P. Vaidyanathan, Multirate systems and Filter Banks, Prentice Hall,1993.

[18]. D. Y. Huang and R. Ma, “Integer fast modified cosine transformation,” in Proc. IEEE Multimedia and Expo, ICME '03. Proceedings pp. 729-732, Oct.2003.

[19]. Z. Zhijin, and Q. Huisheng, “Recursive algorithm for discrete cosine transform,” in Proc. IEEE ICSP, Beijing vol. 1, pp. 115-118, Oct. 1996.

[20]. S. W. Huang, T. H. Tsai, “Fast Decomposition of Filterbanks for theState-of-the-Art Audio Coding”, Signal Processing Letters, IEEE,2005.

[21]. V. Britanak, “On the discrete cosine transform computation,” Signal Processing 40, pp. 183-194, 1994.

[22]. C. W. Kok, “Fast algorithm for computing discrete cosine transform,” IEEE Trans. Signal Processing vol. 45, pp.757-760, Mar. 1997.

[23]. K. R. Rao., P. Yip, Discrete Cosine Transform – Algorithms, Advantages, Applications, Academic Press, Inc., pp. 61-62, 1990.

[24]. Hammami, O., “NOC-based MPSoC design and implementation on FPGA: DCT application” , Quality Electronic Design (ASQED),Jul. 2012.

[25]. J. F. Yang and C. P. Fan, “Recursive Discrete Cosine Transforms with Selectable Fixed-Coefficient Filters”. Transactions on circuit and systems-II: analog and digital signal processing, 1999.

[26]. S. F. Lei, S. C. Lai, P. Y. Cheng, and C. H. Luo, EEE, “Low Complexity and Fast Computation for Recursive MDCT and IMDCT Algorithms,” IEEE Transactions on circuit and systems-II: express briefs, 1996.

[27]. H. C. Chiang, J. C. Liu, “Regressive implementations for the forward and inverse MDCT in MPEG audio coding”, IEEE Signal Processing Letters, vol. 3, is. 4, pp.116-118, Mar. 1996.

[28]. V. Nikolajevic, G. Fettweis, “Computation of forward and inverse MDCT using Clenshaw‘s recurrence formula”, IEEE Transactions on Signal Processing, vol. 51, is. 5, pp. 1439-1444, May 2003.

[29]. C. H. Chen, B. D. Liu and J. F. Yang,” Recursive architectures for realizing modified discrete cosine transform and its inverse,” IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, vol. 50, is. 1, pp.38-45, Jan. 2003.

[30]. S. F. Lei, S. C. Lai, Y. T. Hwang, C. H. Luo, "A high-precision algorithm for the forward and inverse MDCT using the unified recursive architecture," Consumer Electronics, 2008. ISCE 2008. IEEE International Symposium on , vol. 1, no. 4, pp. 14-16, Apr. 2008.