AN MDCT-BASED PSYCHOACOUSTIC MODEL CO-PROCESSOR DESIGN FOR MPEG-2/4 AAC AUDIO ENCODER

Tsung-Han Tsai, Yi-Wen Wang

Dept. of Electrical Engineering
National Central University
Chung-Li, Taiwan, ROC
{han|yiwen}@dsp.ee.ncu.edu.tw

Shih-Way Huang

DSP/IC Design Lab, Grad. Inst. of Elect. Eng.,
Dept. of Electrical Engineering
National Taiwan University, Taiwan, ROC
shihway@video.ee.ntu.edu.tw

ABSTRACT

The Psychoacoustic Model (PAM) is a very important role in MPEG-2/4 Advanced Audio Coding (AAC) encoding. It determines sound quality of a given encoder and influences a lot in computational complexity. This paper presents a new architecture design for MDCT-based PAM co-processor. This work is based on the dedicated hardware design which is different from traditional programmable approaches. Moreover, to reduce the complexity, we replace the calculations of spreading function with reduced fixed coefficients, and decrease the transform kernels from three to one unit.

1. INTRODUCTION

The AAC is an international standard, which is first created in MPEG-2 AAC (ISO/IEC 13818-7), and then become the base of MPEG-4 general audio coding [1]. It is applicable for a wide range of applications from Internet audio to digital audio broadcasting. It also achieves high compression ratio and makes high quality performance due to an improved time-frequency mapping as well as some new coding tools.

The original audio encoding process is in Figure 1. It comprises PAM, Modified Discrete Cosine Transform (MDCT), Spectral Processing (SPP) and Quantization loop (Q loop). PAM calculates a masking threshold, which is the maximum distortion energy masked by the signal energy for each coding partition. Meantime, MDCT transforms input audio samples in time domain into spectrums in frequency domain. The frequency spectrums then transfer to SPP, which removes their redundancies by Temporal Noise Shaping (TNS) and joint coding. Afterward, the spectrums are non-uniformly quantized based on the masking threshold and available number of bits to minimize audible quantization error and then noiselessly coded in the Q Loop. While PAM calculates masking thresholds, SPP and Q loop processes at the same time. Because PAM is deterministic, a dedicated architecture can be implemented. Besides, SPP and Q loop are non-deterministic and a programmable architecture could be an approach for implementation.

Due to the complex calculation and high barrier domain knowledge in PAM, traditional implementations for PAM are mostly direct implementation and porting on high performance programmable approaches such as DSP or general purpose processors [2]. Our previous work has proposed a MDCT-based PAM algorithm with several advantages [3]. In this paper, we propose a dedicated hardware for MDCT-based PAM. According to our architecture, we can have the main advantage of MDCT-based algorithm to replace the complex-FFT computations with MDCT computations. Furthermore, the MDCT kernel used in analysis subband in encoding can be combined with the same MDCT hardware used in PAM. Thus only one transform kernels can be achieved compared with original standardized three kernels approach, one for analysis subband and two for PAM. Therefore, a dedicated architecture of PAM is achieved and can be applied as a co-processor or intellectual property (IP) in system-on-chip (SOC) design.

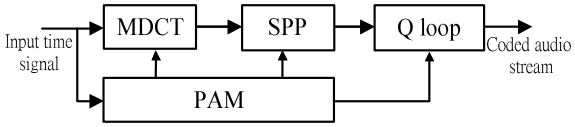

Figure 1: *Block diagram of MPEG-2/4 AAC encoder.*

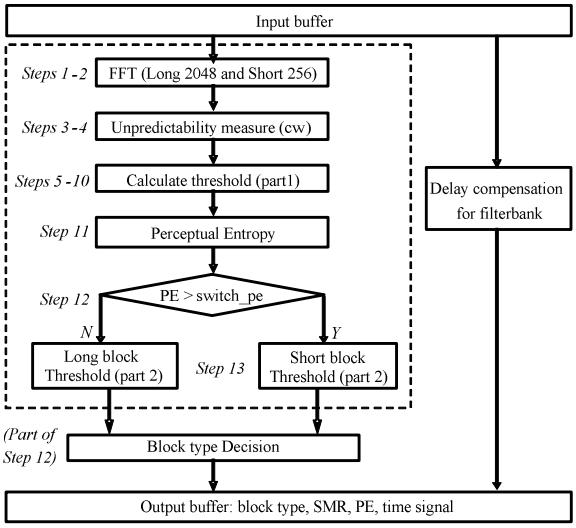

Figure 2: *Block diagram of original PAM.*

2. ALGORITHMS OF MDCT-BASED PAM

In Figure 2, we show the PAM algorithms for simplicity based on the 13 steps [2]. In steps 1-2, PAM normalizes the time-domain samples as input and transforms them into frequency-domain spectrum of real part *r(w)* and imaginary part *i(w)* by FFT. Therefore, when real-part spectrums result in the calculation of partitioned energy, imaginary-part spectrums give rise to the calculation of the unpredictability measure *c(w)* in steps 3-4. In step 5, both partitioned energy and unpredictability are convolved with the spreading

function in order to estimate the effects across the partitioned bands. Tonality index is calculated in step 6 to determine if a signal is tonal-like. Signal-to-Noise Ratio (SNR) is calculated in step 7 and then the masking partitioned energy threshold *nb(b)* is estimated in steps 8-10. Hence, the masking curve is calculated. Perceptual Entropy (PE) is calculated in steps 11-12 to determine the block type. Finally, Signal-to-Masking Ratio (SMR) is estimated in step 13 as output. The *w*, *b*, and *n* indicate indices in the spectral line domain, the threshold calculation partition domain, and the coder scalefactor band domain respectively.

In [3], the MDCT-based PAM is presented. One of the contributions is the reduction on spreading function calculation. In step 5, the calculation of spreading function is composed of a series of complex functions such as comparisons, square roots, power of ten, squares, and divisions. Because the calculation of the spreading function are only dependent by the sampling rates and block type used. Therefore, it can be replaced with look-up tables. The other important contribution is the implementation on MDCT kernel instead of the original FFT kernel. In steps 2-4, there are some special functions such as arctangent, sine/cosine, square root, etc. They have high computational complexity and are hard to be implemented. MDCT-based flow replaced the steps 2-6 with both MDCT and Spectral Flatness Measure (SFM), which are easier to be implemented. According to the analyses, the two mentioned methods can reduce the computational complexity by 80%.

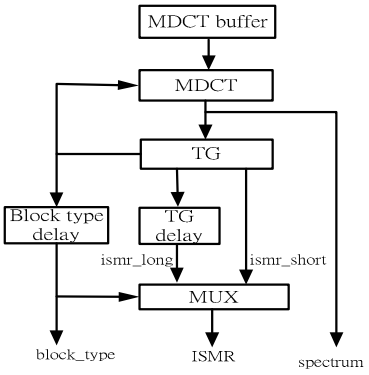

Figure 3: *Block diagram of MDCT-based PAM*

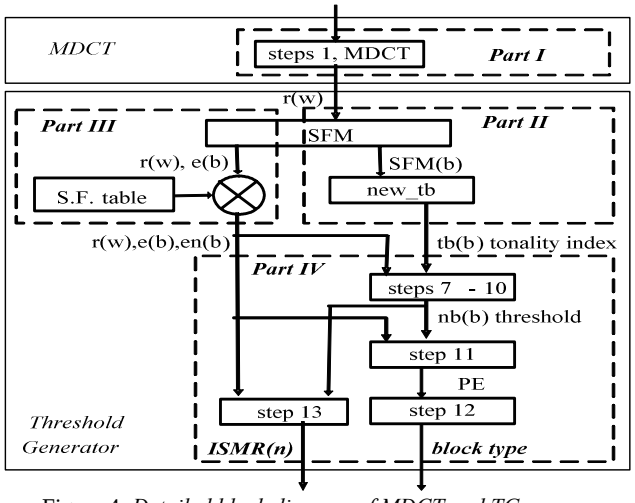

Figure 4: *Detailed block diagram of MDCT and TG.*

3. ARCHITECTURE DESIGN OF MDCT-BASED PAM

The overall MDCT-based PAM is described in Figure 3. In the original approach of standard, there are two FFT in PAM. One is used to calculate FFT with Long block type (2048 points), and the other is used to calculate FFT with Short block type (256 points). Now we replaced Long FFT and Short FFT with MDCT. MDCT should calculate one of the four types according to the block type. Because only one MDCT unit is employed, MDCT and Threshold Generator (TG) require calculating twice a frame. The first time is to estimate the block type and the second time is to calculate the output spectrums. However, TG calculates only once and the Inverse-Signal-to-Masking Ratio (ISMR) is stored in TG delay when the block type is not Short in order to reduce the computational complexity.

Detailed block diagram of MDCT and TG is illustrated in Figure 4. It is decomposed into four parts: Part I consisting of MDCT, Part II consisting of *SFM and* corresponding calculation of tonality *(new_tb)*, Part III consisting of Spreading Function, and Part IV consisting of steps 7-13. TG comprises Part II, Part III, and Part IV. When compared to Figure 2, Figure 4 has some modification. Because MDCT replaces steps 2-4, there is no phase-information. Only the partitioned energies are convolved with the spreading function and SFM is used to generate the tonality index (*new_tb*) from the MDCT spectrums.

The Table I shows specification for input and output data types of MDCT-based PAM. According to Figure 4, the overall architecture design is shown in Figure 5. It also shows the details for the word length of the sign, integer, and fraction part in each block.

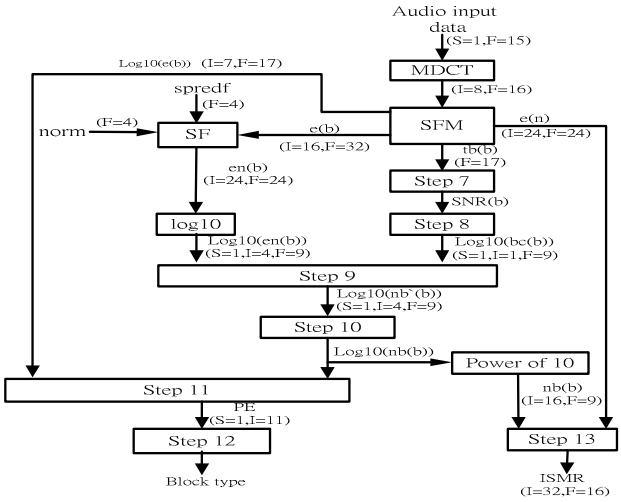

Figure 5: *Overall architecture of MDCT and TG.*

Table I: *Specification of input and output data types (frame).*

Input	Sample rate：44.1kHz Frame size：1024x16b	
Output	Long	Short
	ISMR：49x48b	ISMR：14x48b
	Block type：Long、Start、Stop、Short Spectrum：1024x24b for Long、Start、Stop 128x8x24b for Short	

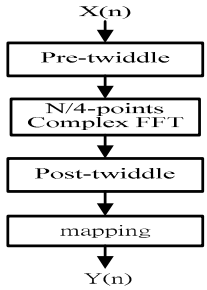

Figure 6: *The flow of MDCT algorithm.*

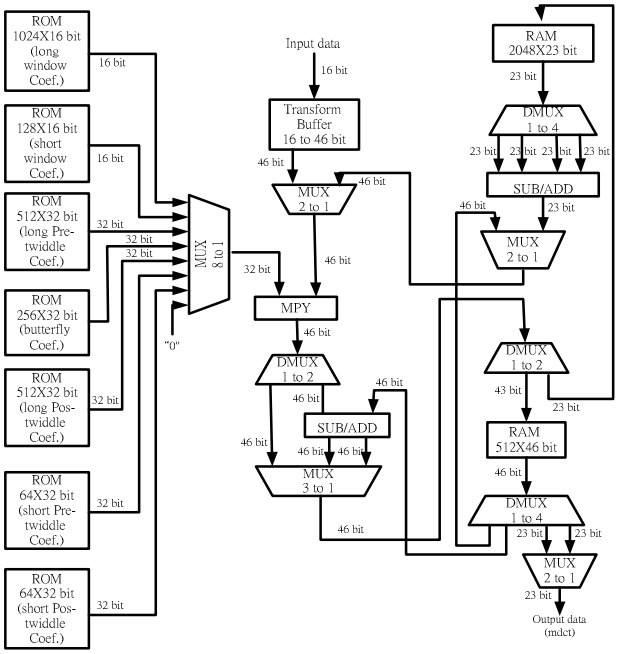

Figure 7: *Overall architecture of MDCT.*

3.1. MDCT

The FFT-based fast algorithm in [4] is adopted for the implementation of MDCT because it can reduce large computational complexity. The processing flow of the fast algorithm is shown in Figure 6. First, the multiplication by pre-twiddle coefficients is applied. Then, the N/4-points complex FFT transform is calculated. In FFT, we use radix-2 butterfly for implementation because of its regularity for dedicated circuits design. N=2048 is for Long type, and N=256 is for Short type. After the N/4-points complex FFT block, 512-points real and imaginary part for Long type, and 64-points real and imaginary part for Short type are obtained respectively. Then they are multiplied by post-twiddle coefficients. Finally, 1024-points real and imaginary part for Long type and 128-points for Short type are obtained as output Y(n).

The implementation of MDCT is shown in Figure 7. The MDCT is implemented and configured as four types, including Long type, Start type, Stop type, and Short type. The main architecture performs multiplication, addition and subtraction to complete MDCT algorithm. The block of MPY involves four multiplications, one addition and one subtraction to finish a complex operation. On the left of Figure 7, those ROM blocks save the coefficients for pre-twiddle, butterfly, and post-twiddle.

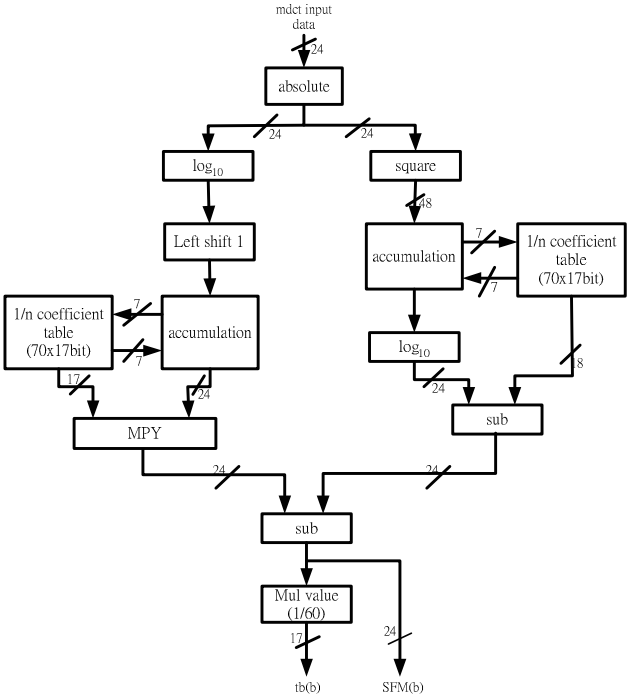

Figure 8: *Overall architecture of SFM and new_ tb.*

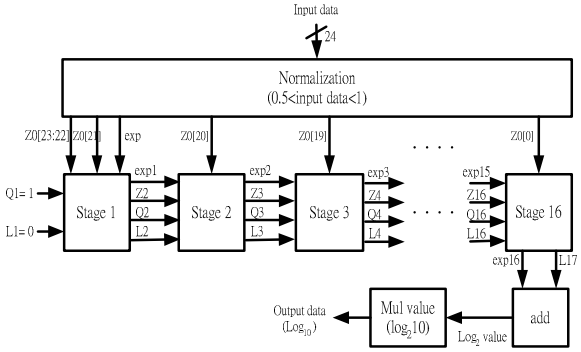

Figure 9: *Block diagram of log_{10}.*

3.2. SFM and new tb

SFM and tb algorithms are defined as follows:

- $$SFM = 10 \log_{10} \frac{Gm}{Am} = 10 \log_{10} \frac{\sqrt[b]{P(1)P(2)....P(b)}}{\frac{P(1)+P(2)+...P(b)}{b}} = 10\left\{\frac{\log_{10}P(1)+\log_{10}P(2)+...+\log_{10}P(b)}{b} - \left[\log_{10}\left(P(1)+P(2)+...P(b)\right)-\log_{10}b\right]\right\} \quad (1)$$
- $$new\_tb = \frac{SFM}{-60} \quad (2)$$

Table II: *The hardware cost of MDCT-Based PAM Co-processor.*

	MDCT buffer		MDCT & TG	
RAM	Numbr	Size	Numbr	Size
	4	1024x16b	1	2048x23b
			1	512x23b
ROM	--		Numbr	Size
	--		1	1024x16b
			2	512x32b
			1	256x32b
			1	128x16b
			2	64x32b
MUL	--		7	
ADD	--		38	

The implementation of *SFM* and *new_tb* is shown in Figure 8. Except some basic computation blocks, Log block is the key module in *SFM* and *new_tb*. Unlike general DSP approach which takes many cycles to complete a Log operation, we use a pipelined architecture for Log implementation with throughput one cycle. According to [5], we can use 16 stages to approximate Log value. The related $\log_{10}$ architecture design is shown in Figure 9. There are some correlations between the stages. The final stage L_{17} will determine Log value.

3.3. Spreading Function (SF)

Spreading function algorithm is defined as:

$$\bullet\ en(b) = nrom\ *(\sum e(b) \otimes sprdngf\ \ (b)) \qquad (3)$$

where $\otimes$ is a convolution operator.

Originally, SF coefficients table is composed of a 70x70 matrices (ex. LONG type). We find the coefficients along the diagonal are non-zeroes and the others are all zeroes except bottom right part. Based on this property, word length is reduced largely. Accordingly, the size of the SF table can be reduced by over 90%.

3.4. Steps 7-13

Steps 7-13 algorithms are described in details as follows:

$$\bullet\ SNR(b) = tb(b)*12 + 6 \quad \ldots\text{step 7} \qquad (4)$$

$$\bullet\ \log\ bc\ (c) = -SNR\ (b)/10 \quad \ldots\text{step 8} \qquad (5)$$

$$\bullet\ \log n`b(b) = \log en(b) + \log bc(b)\ \ \ldots\text{step 9} \qquad (6)$$

$$\bullet\ \log nb(b) = \max(\log qsthr(b), \min(\log n`b(b), \log nb\_1(b) + \log(replev)),$$ where rpelev is 1 for SHORT and 2 for LONG …step 10 (7)

$$\bullet\ PE = PE - (BW*(\log nb(b) - \log e(b))) \ \ldots\text{step 11} \qquad (8)$$

• Block Type Decision. …step 12 (9)

$$\bullet\ SMR = \frac{\sum r(w)^2}{\min(thr(w\_low)...(w\_high))*BW} \ \ldots\text{step 13} \qquad (10)$$

There are several complicated operations such as power, log, and multiplication in the original steps 7-13 of [1]. Referring to [6], which use Log scaling to reduce word length of data such as thresholds, we modify the flow of log and reuse the techniques in Section 3.3. Note that the logarithmic thresholds are converted by power to calculate SMR in Step 13. We also used the pipelined architecture in Log and Power calculation.

4. RESULTS

The hardware cost of the whole MDCT-Based PAM is estimated in Table II. MUL and ADD stands for the number of processing units of multipliers and adders. Most of the hardware is utilized by memory. In our experiments, we simulated the proposed design by Verilog with sampling rate 44100 Hz, it requires 36248 processing cycles per frame. Therefore, the co-processor can be operated about the clock rate 2 MHz to encode stereo channels in real-time.

5. CONCLUSIONS

In this paper, we present a dedicated hardware design of MDCT-based PAM co-processor in MPEG-2/4 AAC encoder. Unlike the traditional programmable approaches, four parts of PAM are optimized to reduce the hardware complexity. This approach can save the transform kernels for three to one unit. Besides, Spreading function is calculated with a reduced size lookup. We also use the pipelined and parallel architectures to improve the performance.

6. REFERENCES

[1] ISO/IEC JTC1/SC29 WG11 No.1650, "IS 13818-7 (MPEG-2 Advanced Audio Coding, AAC)," April 1997.

[2] Yuichiro Takamizawa, Toshiyuki Nomura, Masao Ikekawa, "High-quality and processor-efficient implementation of an MPEG-2 AAC encoder," *in Proceedings of the 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 985–988, 2001.

[3] Tsung-Han Tsai, Shih-Way Huang, Liang-Gee Chen, "Design of a low power psychoacoustic model co-processor for MPEG-2/4 AAC LC stereo encoder," *in Proceedings of the 2003 IEEE International Symposium on Circuits and Systems*, vol. 2, pp. 552–555, 2003.

[4] P. Duhmel, Y. Mahieux, J. P. Petit, "A fast algorithm for the implementation of filter banks based on time domain aliasing cancellation," *International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 2209–2212, Apr. 1991.

[5] Chichyang Chen, Rui-Lin Chen, Chih-Huan Yang, "Pipelined Computation of Very Large Word-Length LNS Addition/Subtraction with Polynomial Hardware Cost," *IEEE Transactions on Computers*, vol. 49, no. 7, pp. 716–726, July 2000.

[6] Marc Gayer, Markus Lohwasser, Manfred Lutzky, "Implementation MPEG Advanced Audio Coding and Layer-3 encoders on 32-bit and 16-bit fixed-point processors," *115th AES Convention,* New York 2003, Preprint 5978.