Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2007, Article ID 89264, 10 pages
doi:10.1155/2007/89264

Research Article

Efficient Algorithm and Architecture of Critical-Band Transform for Low-Power Speech Applications

Chao Wang[1,2] and Woon-Seng Gan[2]

[1] *Center for Signal Processing, School of Electrical and Electronic Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798*
[2] *Digital Signal Processing Lab, School of Electrical and Electronic Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798*

Received 15 December 2005; Revised 8 December 2006; Accepted 18 January 2007

Recommended by Hugo Van Hamme

An efficient algorithm and its corresponding VLSI architecture for the critical-band transform (CBT) are developed to approximate the critical-band filtering of the human ear. The CBT consists of a constant-bandwidth transform in the lower frequency range and a Brown constant-Q transform (CQT) in the higher frequency range. The corresponding VLSI architecture is proposed to achieve significant power efficiency by reducing the computational complexity, using pipeline and parallel processing, and applying the supply voltage scaling technique. A 21-band Bark scale CBT processor with a sampling rate of 16 kHz is designed and simulated. Simulation results verify its suitability for performing short-time spectral analysis on speech. It has a better fitting on the human ear critical-band analysis, significantly fewer computations, and therefore is more energy-efficient than other methods. With a 0.35 μm CMOS technology, it calculates a 160-point speech in 4.99 milliseconds at 234 kHz. The power dissipation is 15.6 μW at 1.1 V. It achieves 82.1% power reduction as compared to a benchmark 256-point FFT processor.

1. INTRODUCTION

Spectral analysis is one of the most fundamental operations in the field of acoustic and speech signal processing. It transforms the time-domain acoustic signal into a frequency-domain spectrum. Some traditional methods, such as fast Fourier transform (FFT), short-time Fourier transform, and filterbank (a group of bandpass filters), have been widely used in academia and industry. These methods usually have a constant frequency resolution. However, psychoacoustical studies show that the human ear performs spectral analysis on the acoustic signal in the form of a filterbank with nonuniform critical bandwidths [1]. For wide-band speech with a bandwidth of 8 kHz, there are 21 critical bands for the Bark scale described by Zwicker [2] and 24 bands for the Mel scale [3]. An interesting finding is that, the bandwidths of the critical bands with center frequencies below a certain frequency are approximately constant. The bandwidths are around 100 Hz below 500 Hz in the Bark scale and below 1 kHz in the Mel scale. Above 500 Hz in the Bark scale or 1 kHz in the Mel scale, the bandwidths increase as the center frequencies increase, while the Q factors of these bandpass filters are approximately constant. Motivated by the human auditory perception model, many methods have been developed to approximate the critical-band analysis. These methods provide advantages over other traditional ways in speech applications, especially in the fields of speech recognition, speech coding, and speech enhancement.

In the past two decades, various schemes to implement critical-band analysis [4–10] have been proposed for speech applications. These methods can be classified into four main approaches: (i) direct digital implementation of the critical-band filterbank, (ii) FFT method, (iii) constant-Q transform (CQT) method, and (iv) wavelet packet transform (WPT) method. The direct implementation of the critical-band filterbank provides good results in the application of speech recognition [4]. In the FFT method, the spectral magnitude of each critical band is obtained by calculating the weighted sum of the FFT magnitude coefficients within the critical band in questions. However, this method requires extra postprocessing in the FFT spectrum. Some typical applications of the FFT method include audio coding [5] and

speech recognition [6]. One of the CQT methods [7] uses constant-Q filters to approximate the critical-band filtering in the high frequency range. In the lower frequency range, the constant-bandwidth coefficients are obtained by summing the constant-Q filters coefficients within each constant-bandwidth band in question. The CQT method in [8] employs the chirp z-transform to approximate the critical-band filtering in the higher frequency range. It uses the FFT to compute the constant-bandwidth coefficients in the lower frequency range. The above methods give a close approximation to the critical-band scale but they are computationally expensive and involve complex hardware architectures. A new approach based on the fast orthogonal WPT (OWPT) was proposed for the applications of speech coding, speech enhancement, and speech recognition [9, 10]. This method uses a tree structure to decompose the input speech signal into the approximated critical bands. However, the disadvantages are the high hardware complexity, and inaccurate approximation to the critical-band scale.

Recently, low-power VLSI speech systems, such as speech recognizers and speech codecs, have many promising applications in large volume battery powered portable products, such as personal digital assistants, communicators and smart toys. The front-end spectral analysis in speech applications, such as the FFT, filterbank and critical-band analysis methods, is both computation intensive and memory intensive, which may consume significant power [11]. The existing CBT methods are not suitable for low-power VLSI realization because of the high computation complexity and high hardware complexity. Therefore, there is a need to design an efficient spectral analyzer for low-power speech systems.

In this study, we develop an efficient critical-band transform algorithm and an architecture for approximating the critical-band filtering of the human ear [12]. The novel CBT scheme has a smaller on-chip memory requirement than the other methods. It also needs fewer computations and less memory access. The proposed VLSI architecture uses a parallel and pipeline structure to increase the throughput. Therefore, a lower supply voltage and a slower clock frequency can be used to achieve significant power reduction.

The remainder of the paper is divided into five sections. Section 2 describes the critical-band transform algorithm. Section 3 presents the short-time spectral analysis of two typical speech phonemes by a 21-band Bark scale CBT. The VLSI architecture and circuit design are presented in Section 4. We evaluate the efficiency of the architecture by designing and simulating the 21-band CBT processor [13], and comparing it against a benchmark 256-point FFT processor we designed. In Section 5, circuit simulation results are reported and discussed. Finally, conclusions are given in Section 6.

2. THE PROPOSED ALGORITHM OF THE CRITICAL-BAND TRANSFORM

Based on the observation of the critical-band scale depicted in Section 1, a novel critical-band transform algorithm is proposed to approximate the critical-band filtering of the human ear. It consists of two transforms: a constant-Q transform (CQT) in the higher frequency range and a constant-bandwidth transform (CBWT) in the lower frequency range. In this study, the Bark scale is approximated.

The Brown CQT algorithm [14] is employed in the proposed CBT. The results in this study show that the Brown CQT with low Q values is a suitable algorithm for speech signal processing. The Brown CQT is also more efficient than the other constant-Q analysis methods. From the discrete short-time Fourier transform, Brown derived an efficient constant-Q transform with a constant ratio of center frequency to frequency resolution (Q). It is known that the resolution Δf of the DFT is equal to the sampling rate divided by the window size (the number of samples analyzed in the time domain). In order to achieve a constant Q, the window size in the Brown CQT varies inversely with frequency. The frequency resolution decreases while the center frequency increases. By choosing a suitable Q value, Brown CQT can achieve a close fitting to the critical bandwidths in the higher frequency range.

The CBWT in the proposed CBT is implemented by using the Brown CQT with a constant window length. The CBWT is formally expressed as

$$X[k_{cw}] = \frac{1}{N_c}\sum_{n=0}^{N_c-1} w[n]x[n] \times \exp\left\{-j2\pi Q_{k_{cw}}\frac{n}{N_c}\right\}, \quad k_{cw} = 1, 2, \ldots, n_{cw}. \tag{1}$$

The window size N_c in the CBWT is constant, while the window size varies for different bands in the original Brown CQT. However, the Q value, $Q_{k_{cw}}$ in the Brown implementation of the CBWT is not constant. The $Q_{k_{cw}}$ is different for n_{cw} constant bandwidths of the CBWT.

In the CBWT, the window size is equal to the sampling rate SR divided by the frequency resolution of 100 Hz,

$$N_c = \frac{\mathrm{SR}}{\Delta f_c} = \frac{\mathrm{SR}}{f_{k_{cw}}}Q_{k_{cw}} = \mathrm{const}. \tag{2}$$

In accordance with the Brown CQT, the CBWT is normalized by dividing it by N_c. The center frequency $f_{k_{cw}}$ of the k_{cw}th spectral component varies linearly with k_{cw}, and is given as

$$f_{k_{cw}} = f_{\mathrm{minc}} + \Delta f_c(k_{cw} - 1), \tag{3}$$

where f_{minc} is the minimum center frequency in the lower frequency range. The center frequency in the Brown CQT is exponential in k_{cq}.

As both the CQT and CBWT in the CBT can be expressed in the Brown CQT form, the proposed CBT is expressed as follows:

$$X[k_{cb}] = \begin{cases} X[k_{cw}], & k_{cb} = k_{cw} = 1, 2, 3, \ldots, n_{cw}; \\ X[k_{cq}], & k_{cb} = k_{cq} = n_{cw} + 1, \ldots, n_{cw} + n_{cq}, \end{cases} \tag{4}$$

where n_{cw}, n_{cq} are the numbers of critical bands in the lower and higher ranges, respectively. The CBT covering the whole

Table 1: Comparison of the parameters in CBWT and CQT.

CBT	CBWT	CQT
Range	Low frequency range of CBT	High frequency range of CBT
Frequency	$f_{\text{minc}} + (k_{cw} - 1)\Delta f_c$ Linear in k_{cw}	$(2^{1/s})^{[k_{cq}-(n_{cw}+1)]} f_{\text{minq}}$ exponential in k_{cq}
Window size	N_c (constant)	$N[k_{cq}] = \text{SR} \times Q_{cq}/f_{k_{cq}}$ (variable)
Bandwidth	SR/N_c (constant)	$f_{k_{cq}}/Q_{cq}$ (variable)
Ratio of frequency to bandwidth	$Q_{k_{cw}}$ (variable)	Q_{cq} (constant)

frequency range can be rearranged into one equation as

$$X[k_{cb}] = \frac{1}{N[k_{cb}]} \sum_{n=0}^{N[k_{cb}]-1} w[k_{cb}, n]x[n] \times \exp\left\{ -j2\pi Q_{k_{cb}} \frac{n}{N[k_{cb}]} \right\}, \quad k_{cb} = 1, 2, \ldots, n_{cw} + n_{cq}, \tag{5}$$

where $X[k_{cb}]$ is the k_{cb}th spectral component of the CBT. Here, $x[n]$ is the discrete-time input speech signal and $w[k_{cb}, n]$ is a window function for each critical band. The length of each window is $N[k_{cb}]$.

The fixed bandwidth in the low frequency range and constant-Q bandwidths in the higher frequency range are defined as

$$\Delta f_{k_{cb}} = \begin{cases} \Delta f_c = 100, & k_{cb} = 1, 2, \ldots, n_{cw}; \\ (2^{1/s})^{[k_{cb}-(n_{cw}+1)]} \times \Delta f_{\text{minq}}, & \\ & k_{cb} = n_{cw} + 1, \ldots, n_{cw} + n_{cq}, \end{cases} \tag{6}$$

where s is the number of constant-Q bands per octave. The k_{cb}th center frequency is expressed as

$$f_{k_{cb}} = \begin{cases} f_{\text{minc}} + \Delta f_c (k_{cb} - 1) = 50 + 100 \times (k_{cb} - 1), & \\ & k_{cb} = 1, 2, \ldots, n_{cw}; \\ (2^{1/s})^{[k_{cb}-(n_{cw}+1)]} \times f_{\text{minq}}, & \\ & k_{cb} = n_{cw} + 1, \ldots, n_{cw} + n_{cq}. \end{cases} \tag{7}$$

Note that 50 Hz is chosen to be the center frequency of the lowest critical band. f_{minq} and Δf_{minq} are the minimum center frequency and bandwidth in the higher frequency range, respectively.

The Q factor of the CBT, $Q_{k_{cb}}$, is therefore described by

$$Q_{k_{cb}} = \frac{f_{k_{cb}}}{\Delta f_{k_{cb}}} = \begin{cases} \dfrac{f_{k_{cb}}}{100}, & k_{cb} = 1, 2, \ldots, n_{cw}; \\ Q_{cq} = \dfrac{1}{(2^{1/s} - 1)}, & k_{cb} = n_{cw} + 1, \ldots, n_{cw} + n_{cq}. \end{cases} \tag{8}$$

In order to reduce spectral leakage, a Hamming window is chosen as the window function $w[k_{cb}, n]$. The length of each window for each critical band is determined by

$$N[k_{cb}] = \frac{\text{SR}}{\Delta f_{k_{cb}}} = \begin{cases} N_c = \dfrac{\text{SR}}{\Delta f_c} & \\ \quad = \dfrac{\text{SR}}{100}, & k_{cb} = 1, 2, \ldots, n_{cw}; \\ \left(\dfrac{\text{SR}}{f_{k_{cb}}}\right) Q_{k_{cb}}, & k_{cb} = n_{cw} + 1, \ldots, n_{cw} + n_{cq}. \end{cases} \tag{9}$$

A comparison between the various parameters used in the CBWT and CQT is given in Table 1. By combining the Hamming window $w[k_{cb}, n]$ and the exponential part into $\text{kern}[k_{cb}, n]$, we can compute the critical-band spectrum by only multiplications and accumulations directly from the input speech data and the precalculated coefficients in (10):

$$X[k_{cb}] = \sum_{n=0}^{N[k_{cb}]-1} x[n] \left\{ \frac{w[k_{cb}, n]}{N[k_{cb}]} \exp\left\{ -j2\pi Q_{k_{cb}} \frac{n}{N[k_{cb}]} \right\} \right\} = \sum_{n=0}^{N[k_{cb}]-1} x[n] \,\text{kern}\,[k_{cb}, n], \quad k_{cb} = 1, 2, \ldots, n_{cb}. \tag{10}$$

In this paper, a 21-band Bark scale CBT with 5 constant-bandwidth bands (100 Hz), and 16 constant-Q bands (Q = 5.6) is constructed at a sampling rate of 16 kHz. The parameter values are chosen so that the 21-band CBT closely approximates the Bark scale. For the Mel scale, there are 10 constant-bandwidth bands, and 14 constant-Q bands with Q = 6.9.

3. SHORT-TIME CRITICAL-BAND ANALYSIS ON SPEECH

In this section, the performance of the proposed 21-band Bark scale critical-band transform is evaluated and compared with the OWPT method. Figure 1 shows the degree of approximation to the Bark scale critical bands both for the CBT and for the OWPT methods [9]. It shows that the proposed CBT provides a closer approximation to the Bark scale, especially in terms of the bandwidths. This is because the OWPT method can only divide the bandwidths by a factor of 2.

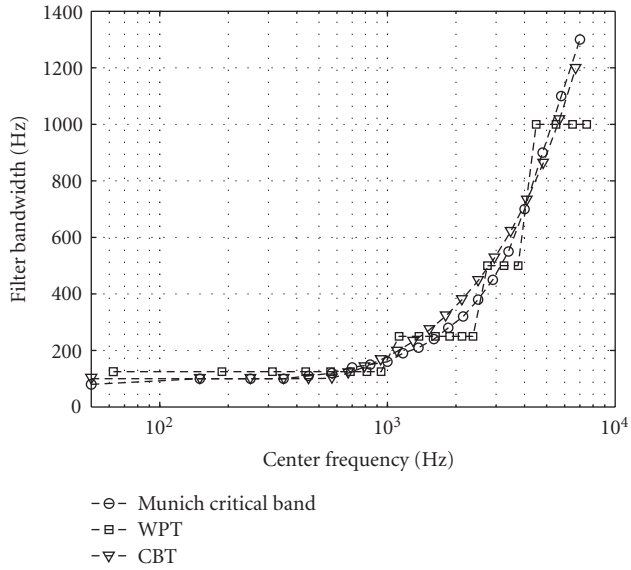

Figure 1: Degree of approximation to Munich Bark critical bands.

The 21-band CBT algorithm has been programmed and simulated in Matlab 6.5. A typical utterance "ka" [8] is used in our testing. The syllable "ka" consists of two 600-ms waveforms for "k" and "a," respectively. The 1200-ms speech spoken by a male talker was recorded in a small room and processed by CoolEdit Pro 2.0 at a sampling rate of 16 kHz. The 21-band CBT uses 1/2-overlap processing on the 160-point segments of the speech. The CBT spectra of the two speech waveforms are shown in Figures 2(a) and 2(b), respectively. The corresponding FFT spectra are given in Figures 3(a) and 3(b), respectively. These plots show the short-time spectral magnitude on the z-axis against the frequency in a log scale on the x-axis. The labels on the y-axis correspond to the speech duration in seconds.

In the first 600 milliseconds in Figure 2(a), the initial burst of energy of the plosive "k" has a concentration of energy in the region near 2 kHz. The energy peak at the very low frequency range is also observed in the FFT spectra as shown in Figure 3(a). It is commonly observed in spectrogram analysis of the speech signal. A clear formant structure for the vowel "a" can be observed from Figure 2(b), with the first and second formant frequencies around 650 Hz and 1100 Hz, respectively. The third formant around 2500 Hz can also be seen. These formant frequencies are the typical features of the vowel "a" [15]. The short-time spectra as shown in Figure 2 for the CBT follow closely those obtained by a 256-point FFT as shown in Figure 3. The proposed CBT is not invertible as the Brown CQT is not invertible [14]. However, it is adequate to show the typical spectral features of the phonemes. In some speech applications, the pitch is ignorable and the higher frequency information is less significant [16]. But the critical-band analysis based on the Bark scale or Mel scale can still capture the phonetically important characteristics of speech. It may work effectively and well in speech recognition [3, 4].

Based on the above analysis and discussion, the proposed 21-band CBT performs spectral analysis of speech satisfactorily. It can be used as an auditory spectral analyzer in speech applications.

4. THE VLSI ARCHITECTURE OF THE CRITICAL-BAND TRANSFORM

In this section, an efficient VLSI architecture is proposed for the critical-band transform. By applying the symmetry property of the CBT coefficients, the number of multiplications is reduced by about 50%. The derived data path can easily be pipelined and parallelized. It is very suitable for an ASIC implementation.

4.1. The VLSI architecture of the critical-band transform

It is observed that there is a symmetry property of the CBT coefficient kern in (10). The coefficient consists of a real part (the cosine function) and an imaginary part (the sine function). Applying the symmetry property of the cosine function and antisymmetry property of the sine function, the CBT can be rearranged as

$$
\begin{aligned}
&X[k_{cb}] \\
&= \sum_{n=0}^{N[k_{cb}]-1} x[n]\left(\cos[k_{cb}, n] + j * \sin[k_{cb}, n]\right) \\
&= \begin{cases}
\sum_{n=1}^{M[k_{cb}]} \{(x[n]+x[N-n])\cos + j*(x[n]-x[N-n])\sin\} \\
\quad +(x[0]+0)\,\mathrm{kern}[0], & N[k_{cb}] \text{ is odd}, \\
\sum_{n=1}^{M[k_{cb}]-1} \{(x[n]+x[N-n])\cos + j*(x[n]-x[N-n])\sin\} \\
\quad +(x[0]+0)\,\mathrm{kern}[0] + (x[M]+0)\,\mathrm{kern}[M], \\
\qquad\qquad N[k_{cb}] \text{ is even},
\end{cases}
\end{aligned}
\tag{11}
$$

where

$$
M[k_{cb}] = \begin{cases}
\dfrac{(N[k_{cb}]-1)}{2}, & N[k_{cb}] \text{ is odd}, \\
\dfrac{N[k_{cb}]}{2}, & N[k_{cb}] \text{ is even}.
\end{cases}
\tag{12}
$$

There are two operation modes for calculating the CBT spectrum of each critical band, when the window length is odd and even, respectively. By inserting zeroes into the equation, we can derive the regular expressions as described by (11). Therefore, the number of multiplications and memory usage are reduced by about 50%. These savings contribute significantly not only to the reduction of the memory area but also to the saving of power consumption by frequent

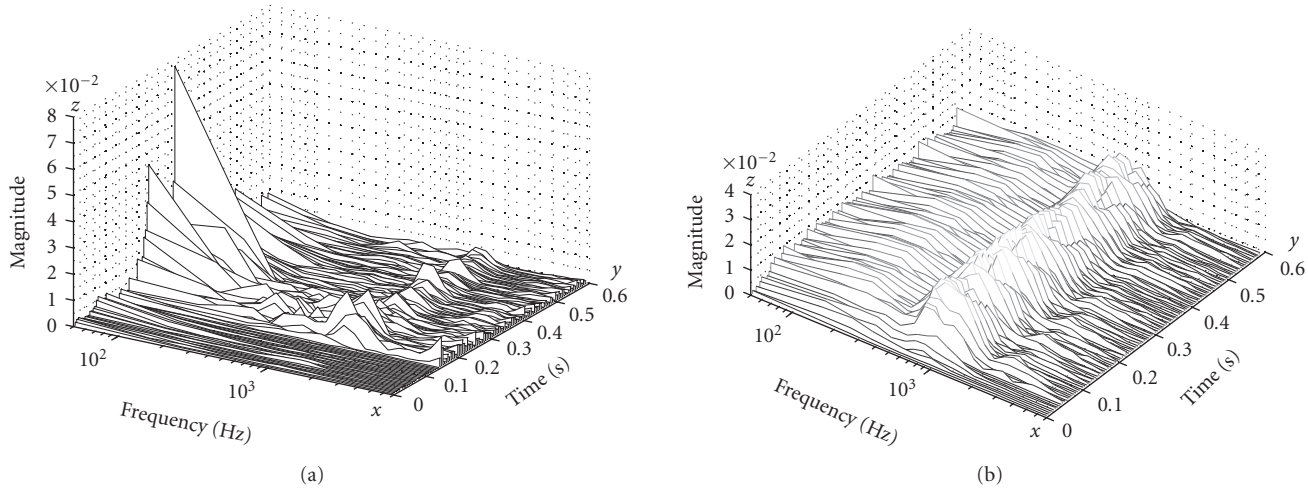

Figure 2: (a) CBT analysis of the first 600 ms of "ka"; (b) CBT analysis of the second 600 ms of "ka."

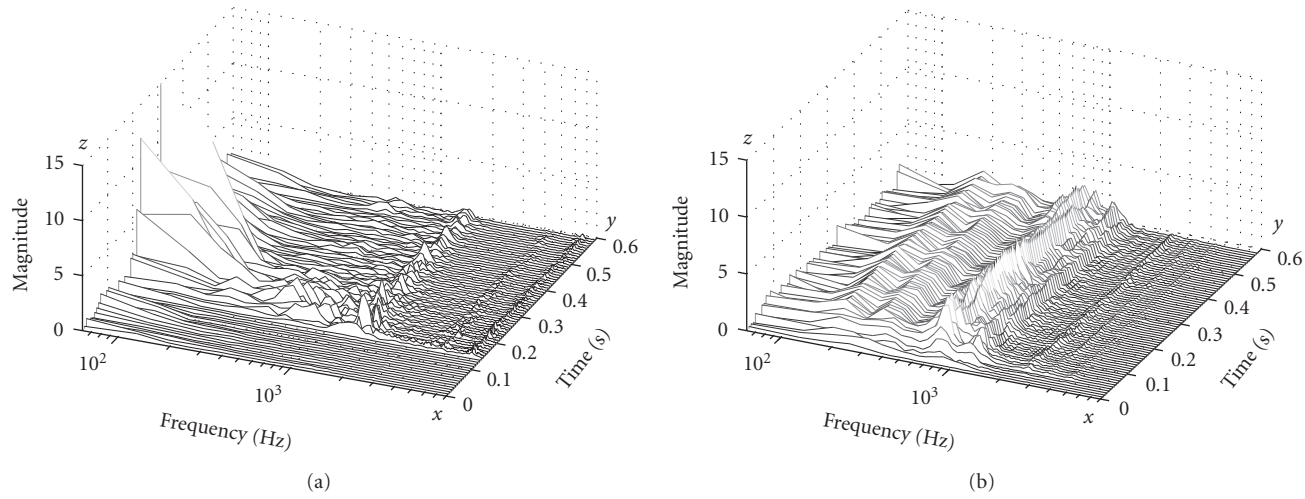

Figure 3: (a) FFT analysis of the first 600 ms of "ka"; (b) FFT analysis of the second 600 ms of "ka."

memory access. The data flow of the CBT is derived from (11). As depicted in Figure 4, the CBT spectral magnitude for each critical band is obtained after all the accumulations over a window of input speech samples complete. We denote the addition (or subtraction) and multiplication-accumulation (MAC) process of a pair of data elements as one butterfly operation.

The proposed VLSI architecture of the critical-band transform processor consists of a pipelined data path, a controller, a coefficient ROM, a data input RAM, a data output RAM, and an address generator. In this study, the I/O data and coefficients are expressed in the 16-bit two's complement fixed-point format. The operation of the processor is partitioned into data I/O process (I/O mode) and CBT computation process (CBT mode).

From the CBT data flow depicted in Figure 4, we propose a two-multiplier and four-adder pipelined data path as shown in Figure 5. The data are processed in two parallel

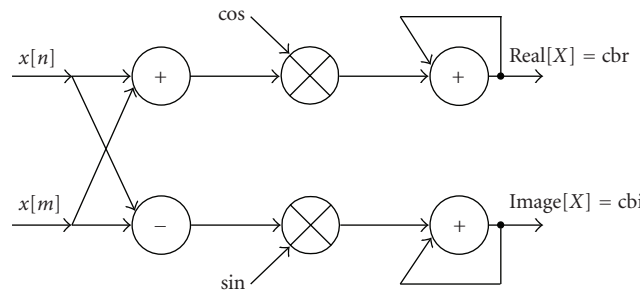

Figure 4: Data flow graph of the CBT algorithm.

paths. The efficient pipeline and parallel processing makes it possible to utilize the supply voltage scaling approach to achieve significant power reduction [17]. It has three pipeline stages to improve the processing throughput. In the first

Table 2: Pipeline table of CBT data path.

RAM read	First 2 adds	Two mults.	Second 2 adds	—
$x[n]$	$x[n]+0$ $x[n]-0$ read kern	$x[n]\times\cos$ $x[n]\times\sin$	2 accumulations	—
—	RAM read	First 2 adds	Two mults.	Second 2 adds
—	$x[n], x[m]$	$x[n]-x[m]$ $x[n]+x[m]$ read kern	$(x[n]-x[m])\times$kern $(x[n]+x[m])\times$kern	2 accumulations

Table 3: Last butterfly operation in the pipeline.

(a) When window size $N[k_{cb}]$ is odd

RAM read	First 2 adds	Two mults.	Second 2 adds	RAM write
$x[n], x[m]$	$x[n]+x[m]$ $x[n]-x[m]$ read kern	$(x[n]-x[m])\times$kern $(x[n]+x[m])\times$kern	2 accumulations	cbr, cbi

(b) When window size $N[k_{cb}]$ is even

RAM read	First 2 adds	Two mults.	Second 2 adds	RAM write
$x[n]$	$x[n]+0$ $x[n]-0$ read kern	$x[n]\times\cos$ $x[n]\times\sin$	2 accumulations	cbr, cbi

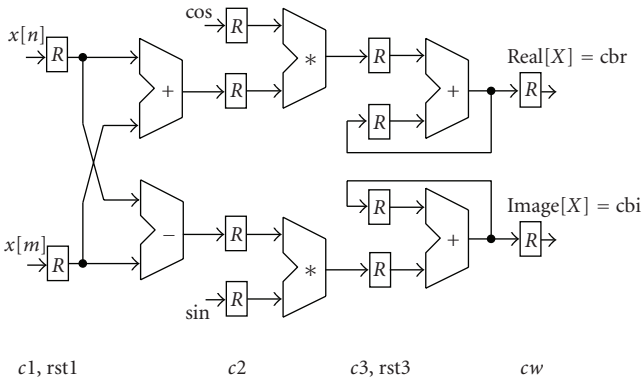

Figure 5: Proposed pipelined CBT data path.

stage, the first pair of 16-bit wide adders processes two data elements from the input RAM. The two multipliers compute 16-bit × 16-bit multiplications and produce 32-bit results for each multiplier in the second stage. In the last stage, the second pair of 32-bit wide adders performs the accumulations. The final results are truncated into 16-bits and written to the output RAM, when a CBT spectrum computation is completed.

As described in (11), for a particular CBT spectrum, there are $(N[k_{cb}]-1)/2+1$ butterfly operations when $N[k_{cb}]$ is odd, or $N[k_{cb}]/2+1$ butterflies when even. The pipeline processing of the butterfly operations is described in Table 2. In the first butterfly operation for each critical band, only one data element is read from the input RAM and fed into one of the first pair of pipeline registers. At the same time, the other register is reset to zero as described in (11). As shown in Table 3, the CBT data path has two working modes, that is, even mode and odd mode. This is because the last butterfly operation might be different for individual critical bands. For the odd window length, a pair of data elements is read from the input RAM as usual but only one data element is read when the window size is even. It takes the data path $(N[k_{cb}]-1)/2+4$ cycles to compute a CBT spectrum (including access of the I/O memories) when $N[k_{cb}]$ is odd, and $N[k_{cb}]/2+4$ cycles when $N[k_{cb}]$ is even.

The proper pipeline processing with the two working modes is controlled by a controller. By multiplexing the data path, CBT spectra are computed one by one from band 1 to band n_{cb}. This controller also supervises the other functional units in the processor for proper operation. The coefficient ROM stores the precomputed CBT coefficients kern, and the I/O RAMs are used to buffer the input speech data and output CBT spectra. Another important functional unit is the address generator, which provides the correct addresses for the I/O RAMs and the coefficient ROM. It consists of the critical-band generator and the address generation unit. The critical-band generator keeps track of which CBT spectrum is being computed. It also provides the controller and the address generation unit with the information of each critical

band, including the number of the butterfly operations, parity of the window size, and the offset values for calculating the correct addresses in the CBT mode. This information has been prestored in the critical-band generator when a particular CBT is determined. The address generation unit generates addresses for the coefficient ROM in CBT mode and for the I/O RAMs in both CBT and I/O modes.

For comparison, we also design a 256-point radix-2 DIT (decimation-in-time) in-place FFT processor based on a single-butterfly architecture, as a benchmark against the proposed CBT processor. The benchmark FFT processor consists of a controller, a coefficient ROM, a data RAM, an address generation unit, and a pipelined butterfly unit with only two multipliers and three adders. The I/O data and coefficients are also represented in the 16-bit two's complement fixed-point format.

The implementation of the butterfly unit is very crucial in the design of a single-butterfly FFT processor. In the literature, there are mainly three methods using different numbers of multipliers and adders to implement the radix-2 DIT butterfly unit. The radix-2 DIT butterfly is described by

$$\begin{aligned} C &= A + W \times B, \\ D &= A - W \times B, \end{aligned} \tag{13}$$

where W is the twiddle factor. In (13), A and B are the two inputs, while C and D are the two outputs. All the variables are complex numbers. By replacing the complex variables with real variables, a fully parallel butterfly structure with four multipliers and six adders in [18] was derived to achieve the highest throughput. The four-multiplier and six-adder butterfly unit computes one butterfly operation every cycle. To reduce the hardware cost, a one-multiplier and two-adder butterfly unit in [19] was proposed to compute one butterfly operation every four cycles by multiplexing just one multiplier and two adders. By considering both performance and cost, the two-multiplier and four-adder implementation provides the best trade off as claimed in [20]. The throughput is two cycles for one butterfly operation, while the control is much simpler.

In the benchmark 256-point FFT processor, we design a two-multiplier and three-adder radix-2 DIT butterfly unit derived from the rewritten butterfly equation (14)

$$\begin{aligned} X &= B_R \times W_R - B_I \times W_I, \\ C_R &= A_R + X, \\ D_R &= A_R - X, \\ Y &= B_I \times W_R + B_R \times W_I, \\ C_I &= A_I + Y, \\ D_I &= A_I - Y. \end{aligned} \tag{14}$$

In (14), the subscripts "R" and "I" are used to denote the real part and imaginary part of the complex variables, respectively. For simplicity, the j prefix associated with the imaginary part is omitted. From (14), a rescheduled SFG for the radix-2 butterfly is derived as shown in Figure 6. Based on the SFG, we propose a two-multiplier and three-adder pipelined butterfly unit as depicted in Figure 7. Compared

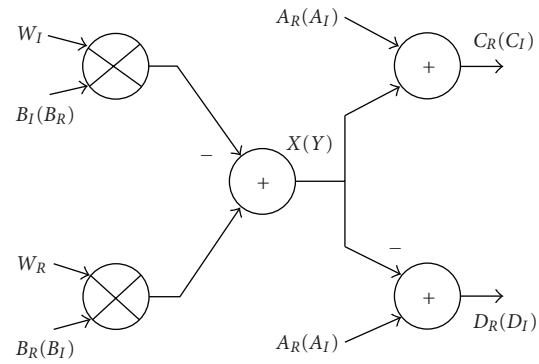

Figure 6: Rescheduled data flow graph for the radix-2 butterfly.

with the two-multiplier and four-adder scheme, it can still achieve a throughput of two cycles with a latency of four cycles, while it has less hardware cost by reducing the number of adders from four to three. It is a good solution with a good trade-off for low-cost speech applications. The proposed two-multiplier and three-adder butterfly unit is employed to compute the butterfly operations recursively in the benchmark FFT processor.

In high-performance applications, such as image, video, and radar signal processing, the pipeline architecture [21] and the parallel architecture [22] using multiple butterfly units are widely used to compute the high-speed long-sized FFT. All these architectures including the single-butterfly methods provide users flexibility to make a trade off between hardware cost and performance, by choosing different numbers of butterfly units to achieve a different throughput for a particular application. However, our study focuses low-cost speech applications. The multiple-butterfly pipeline and parallel architectures are not necessary and too expensive as the performance requirement of speech applications is not high. For example, the array FFT processor designed in [22] uses four butterfly units to compute the FFT. Each butterfly unit consists of two multipliers and four adders. So the hardware cost required by the butterfly units in the array processor is four times that of the single-butterfly architecture. Given the segments of 256-point speech samples at a sampling rate of 16 kHz, the single-butterfly FFT architecture can easily meet the real-time processing requirement. Because of low cost requirements, we chose the single-butterfly architecture to design the benchmark 256-point FFT processor.

4.2. Computation complexity and memory access

Since most of the operations in DSP algorithms involve multiplications and accumulation, the multiplication and addition operations are commonly used to measure the efficiency of DSP algorithms. In this section, the numbers of multiplications and additions are used to evaluate the power-efficiency of the proposed CBT algorithm and architecture.

In the proposed CBT, the number of the complex multiplications is half of the window lengths due to the coefficient symmetry property. The input speech data is always real

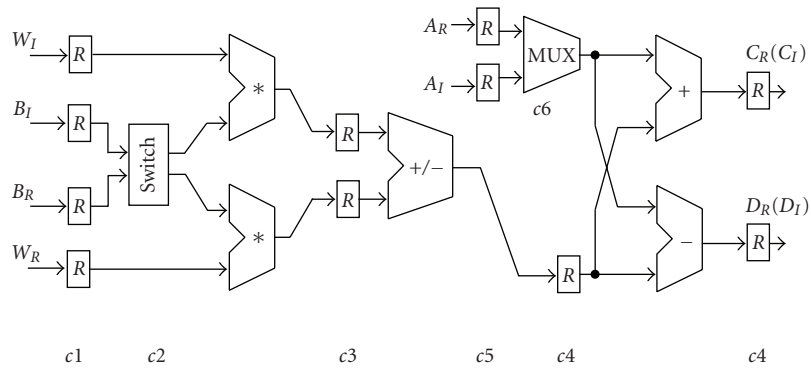

Figure 7: Proposed pipelined radix-2 butterfly unit.

Table 4: Comparison of on-chip memory access.

Auditory spectral analyzer	RAM access			Total memory access
	Input write	R/W during computation	Output read	
256-point in-place FFT processor	256	8192 (512×2 × 8)	512	8960
21-band CBT processor	160	1808 (1766 + 42)	42	2010

and the coefficients are complex. The 21-band CBT involves 1766 real multiplications and 3466 real additions. Both the numbers of real multiplications and real additions in the 256-point FFT are 4096. The OWPT method, using 10-order Daubechies filters, consumes 9216 real multiplications and 3800 real additions in a frame of 64 samples [9]. The number of multiplications in the CBT is 56.9% less than in the FFT, while the saving in the real additions is 15.4%. The reduction as compared to the OWPT is more significant. Recently, the lifting technique is widely used in wavelet transforms to reduce the computation complexity by up to 50% [23]. If the lifting technique is used in the WPT method, the computation is still larger than in the CBT.

In most typical DSP algorithms, frequent memory access is another important contribution to the total power dissipation. Therefore, the memory access of the proposed CBT processor is also compared with that of the 256-point FFT processor in this section. For the proposed 21-band CBT processor, the on-chip memory consists of a 1766-word × 16-bit ROM, a 160-word × 16-bit RAM, and a 42-word × 16-bit output RAM. The 256-point FFT processor requires a 256-word × 16-bit coefficient ROM and a 512-word × 16-bit RAM. The comparison on RAM access is given in Table 4. The CBT requires a total of 2010 read/write RAM accesses. This is in contrast to the 8960 accesses required for a 256-point in-place FFT. The 21-band CBT results in a reduction of 77.6% in memory accesses as compared to the FFT.

5. CIRCUITS SIMULATION RESULTS AND ANALYSIS

The proposed 21-band Bark scale CBT processor and the benchmark 256-point FFT processor are designed by using VHDL. The CBT processor takes 1167 cycles to compute a 21-band CBT. The FFT processor computes a 256-point FFT in 2572 cycles.

Both the CBT processor and the FFT processor are simulated at RTL by using Mentor Graphics Modelsim. They have been synthesized into gate level by the Synopsys design compiler with the AMS 0.35 μm CMOS standard cell library. The estimated areas of the two processors are 2.69 mm^2 and 9.02 mm^2, respectively. The estimated maximum clock frequencies are 83.3 MHz and 100 MHz, respectively. In order to estimate the power dissipation, the two processors are simulated at transistor level by Synopsys Nanosim. Simulation at transistor level shows that the CBT processor can still work at a maximum clock frequency of 13 MHz, when the supply voltage is scaled down to 1.1 V. It can achieve real-time processing at 234 kHz. Table 5 lists the percentage dissipation for the different functional units at 234 kHz and 1.1 V. Table 6 shows the estimated power dissipation at 1.1 V when the clock frequency is 234 kHz and 1 MHz, respectively. The CBT processor operates at 50% overlap on 160-point data segments at a sampling rate of 16 kHz.

Table 5 shows that the multiplications and RAM memory accesses consume the largest portion of the total power dissipation, which is 52.1% and 17.6%, respectively. It is shown in Table 6 that the CBT processor can achieve about 95.3% power saving at 234 kHz by scaling the supply voltage from 3.3 V to 1.1 V.

As a benchmark, the 256-point FFT processor can perform real-time processing within 7.7 milliseconds at 322 kHz and 1.1 V. It operates at 50% overlap on 256-point data segments. The FFT processor consumes 87.1 μW per FFT, while the CBT processor consumes only 15.6 μW per CBT.

Table 5: Power dissipation percentage for different functional units in the CBT processor.

Functional units	Address generator	Controller	I/O RAM	ROM	Data path (multiplications)
Percentage of the total power dissipation	4.6%	2.8%	17.6%	2.9%	71.3% (52.1%)

Table 6: CBT processor power dissipation simulation results under 1.1 V and 3.3 V.

Supply voltage (V)	3.3	1.1
Clock frequency (MHz)	0.234	0.234
Average power (μW/MHz)	1413.6	66.7

6. CONCLUSIONS

An efficient algorithm and its VLSI architecture for the critical-band transform have been proposed for speech applications. Comparative studies were conducted to show that the proposed 21-band Bark scale CBT is better than the OWPT and FFT methods in terms of the closeness in approximation to human ear critical-band filtering, computational complexity, and memory access. Simulation results verified its suitability for performing short-time spectral analysis on speech. Circuits design and simulation of the CBT processor and a benchmark 256-point FFT processor verified the power efficiency of the proposed architecture. The proposed CBT algorithm and its architecture are very suited for low-power speech applications.

REFERENCES

[1] H. Fletcher, "Auditory patterns," *Reviews of Modern Physics*, vol. 12, no. 1, pp. 47–65, 1940.

[2] E. Zwicker, "Subdivision of the audible frequency range into critical bands (frequenzgruppen)," *The Journal of the Acoustical Society of America*, vol. 33, no. 2, p. 248, 1961.

[3] J. W. Picone, "Signal modeling techniques in speech recognition," *Proceedings of the IEEE*, vol. 81, no. 9, pp. 1215–1247, 1993.

[4] B. A. Dautrich, L. R. Rabiner, and T. B. Martin, "On the effects of varying filter bank parameters on isolated word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 31, no. 4, pp. 793–807, 1983.

[5] P. Noll, "Digital audio coding for visual communications," *Proceedings of the IEEE*, vol. 83, no. 6, pp. 925–943, 1995.

[6] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.

[7] T. L. Petersen and S. F. Boll, "Critical band analysis-synthesis," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 31, no. 3, pp. 656–663, 1983.

[8] J. M. Kates, "An auditory spectral analysis model using the chirp z-transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 31, no. 1, pp. 148–156, 1983.

[9] B. Carnero and A. Drygajlo, "Perceptual speech coding and enhancement using frame-synchronized fast wavelet packet transform algorithms," *IEEE Transactions on Signal Processing*, vol. 47, no. 6, pp. 1622–1635, 1999.

[10] O. Farooq and S. Datta, "Mel filter-like admissible wavelet packet structure for speech recognition," *IEEE Signal Processing Letters*, vol. 8, no. 7, pp. 196–198, 2001.

[11] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low power techniques for portable real-time DSP applications," in *Proceedings of the 5th International Conference on VLSI Design*, pp. 203–208, Bangalore, India, January 1992.

[12] C. Wang and Y.-C. Tong, "An improved critical-band transform processor for speech applications," in *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS '04)*, vol. 3, pp. 461–464, Vancouver, BC, Canada, May 2004.

[13] C. Wang, Y.-C. Tong, and Y. Shao, "VLSI design and analysis of a critical-band transform processor for speech recognition," in *Proceedings of IEEE International SOC Conference*, pp. 365–368, Santa Clara, Calif, USA, September 2004.

[14] J. C. Brown, "Calculation of a constant Q spectral transform," *Journal of the Acoustical Society of America*, vol. 89, no. 1, pp. 425–434, 1991.

[15] L. Rabiner and B. Juang, *Fundamentals of Speech Recognition*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1993.

[16] J. N. Holmes and W. J. Holmes, *Speech Synthesis and Recognition*, Taylor & Francis, New York, NY, USA, 2nd edition, 2001.

[17] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low-power CMOS digital design," *IEEE Journal of Solid-State Circuits*, vol. 27, no. 4, pp. 473–484, 1992.

[18] B. M. Bass, "A low-power, high-performance, 1024-points FFT processor," *IEEE Journal of Solid-State Circuits*, vol. 34, no. 3, pp. 380–387, 1999.

[19] E. Cetin, R. C. S. Morling, and I. Kale, "An integrated 256-point complex FFT processor for real-time spectrum analysis and measurement," in *Proceedings of IEEE Instrumentation and Measurement Technology Conference*, vol. 1, pp. 96–101, Ottawa, ON, Canada, May 1997.

[20] P. A. Ruetz and M. M. Cai, "A real time FFT chip set: architectural issues," in *Proceedings of the 10th International Conference on Pattern Recognition*, vol. 2, pp. 385–388, Atlantic City, NJ, USA, June 1990.

[21] E. Bidet, D. Castelain, C. Joanblanq, and P. Senn, "A fast single-chip implementation of 8192 complex point FFT," *IEEE Journal of Solid-State Circuits*, vol. 30, no. 3, pp. 300–305, 1995.

[22] Z. Liu, Y. Song, T. Ikenaga, and S. Goto, "A VLSI array processing oriented fast Fourier transform algorithm and hardware implementation," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 88, no. 12, pp. 3523–3530, 2005.

[23] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," *Journal of Fourier Analysis and Applications*, vol. 4, no. 3, pp. 247–269, 1998.

Chao Wang received his B.Eng. degree in electronics engineering from the Department of Electronics Science and Technology, Huazhong University of Science and Technology, Wuhan, China, in 2000. Currently, he is a Ph.D. Candidate in the School of Electrical and Electronic Engineering, Nanyang Technological University (NTU), Singapore. He is also with the Center for Signal Processing, NTU as a Research Engineer. His research interests include digital IC design, VLSI architectures for digital signal processing, low-power design, and embedded signal processing.

Woon-Seng Gan received his B.Eng. (1st class hons) and Ph.D. degrees, both in electrical and electronic engineering from the University of Strathclyde, UK, in 1989 and 1993, respectively. He joined the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, as a Lecturer and Senior Lecturer in 1993 and 1998, respectively. In 1999, he was promoted to an Associate Professor. He teaches several undergraduate, postgraduate, and industry courses on digital signal processing and real-time signal processing implementation. His research interests include adaptive signal processing, psycho acoustical signal processing, image processing, and real-time digital signal processing. He has published more than 130 international refereed journals and conferences. He has coauthored a book on "*Digital Signal Processors: Architectures, Implementations, and Applications*," Prentice Hall, 2005, and he is the leading author of a latest book on "*Embedded Signal Processing with the Micro Signal Architecture*," Wiley-IEEE Press, 2007.