Sub μW Noise Reduction for CIC Hearing Aids

Cheng-Wen Wei, Sheng-Jie Su, Tian-Sheuan Chang, *Senior Member, IEEE*, and Shyh-Jye Jou, *Senior Member, IEEE*

***Abstract*—This paper presents a sub μW noise reduction design to enhance speech for completely-in-the-canal (CIC) type hearing aids by optimizing its algorithm and associated architecture. In algorithm optimization, a low-complexity mixed perceptual-discrete wavelet packet transform (P-DWPT) and fast Hartley transform (FHT) are adopted for spectral decomposition and reconstruction. A simple yet efficient denoise method with 4-zone-voice activity detection (VAD) supports a consonant protection to improve speech quality and a skip scheme to reduce power consumption. In the designed architecture, mixed P-DWPT and FHT are folded into one 8-by-8 configurable butterfly computation unit with on-time scheduling for low power operation. The circuit is implemented with 0.18-μm CMOS process and consumes only 0.65 μW power at 1.0 V with a speech quality that is comparable to that achieved using other high-complexity algorithms.**

***Index Terms*—Acoustic noise, hearing aids, low power design, speech processing, VLSI.**

I. Introduction

NOISE degrades the quality of speech heard using hearing aids, often reducing a person's willingness to wear such aids. To solve this problem, hearing aids algorithms must incorporate noise reduction. However, since battery power is limited, the power consumed by noise reduction in hearing aids should be minimized while acceptable performance is maintained.

Low-power approaches of noise reduction in hearing aids have two components, i.e., algorithm and architecture. With respect to the algorithm, various noise reduction methods have been proposed, including adaptive noise reduction [1], Wiener filtering, subspace, or spectral subtraction [2]. However, adaptive noise reduction requires two or more microphone inputs and is not suitable for CIC hearing aids (which are the focus of this paper,) which have only one microphone for sound acquisition. Additionally, all of these methods, except for spectral subtraction, are too complicated to be applied in power-limited hearing aid devices. As a result, this investigation uses the spectral subtraction method for algorithm development.

With respect to the implementation, several results concerning the noise reduction in hearing aids have been presented. One result, related to the 0.25 μm process [3], exploits folded architecture and consumed 45 μW at 1.25 V. However, the algorithm of this design is based on an adaptive directional microphone that consists of two microphones. Thus, it is not suitable for CIC hearing aids. Another design based on the 0.35-μm process [4] utilizes parallel (unfolded) architecture and subthreshold pseudo NMOS logic to implement delayed LMS (DLMS) with a power consumption of 44 μW (22 nJ per operation) at 0.4 V. However, the associated algorithm is also for a two-microphone or multi-microphone adaptive filter. In addition, to maintain stable operation in the subthreshold region, several compensation circuits are embedded. Thus, the design is not a cell-based digital design and suffers from performance deviation caused by process, voltage, and temperature (PVT) variations. In addition to application-specific integrated circuit (ASIC), several digital signal processing (DSP) processor-based implementations have been presented to provide noise reduction in hearing aids [5]–[7]. However, they do not dedicatedly consider the complexity of noise reduction and thus consume much power.

This work describes a low power noise reduction design for completely-in-the-canal (CIC) hearing aids by optimizing the algorithm and associated architecture. In the algorithm, a potential low-power spectral subtraction method, which includes frequency decomposition and reconstruction, denoise, and voice activity detection (VAD) is adopted. For decomposition and reconstruction, a novel mixed and perceptual discrete wavelet packet transform (DWPT) and fast Hartley transform (FHT) instead of fast Fourier transform (FFT) is proposed to fit human perception better while reducing hardware complexity. Then, a simple yet efficient de-noise method with 4-zone-VAD and consonant protection is adopted to maintain speech quality. A low-power architecture is achieved without increasing clock by performing decomposition and reconstruction on a single configurable 8-by-8 butterfly unit with on-time scheduling. The power consumed by the reconstruction is further reduced using a skip scheme controlled by VAD. The final implementation consumes only 0.65 uW at 1.0 V using 0.18-μm CMOS technology.

The rest of this paper is organized as follows. Section II describes the details of the proposed algorithm. Section III then shows the simulation results. Next, Section IV describes the architecture design and implementation results are presented in Section V. Conclusions are finally drawn in Section VI.

II. Related Works and Proposed Algorithm

The spectral subtraction method, introduced by Boll [8], comprises two basic steps, which are the spectral decomposition and reconstruction and the denoise process.

Manuscript received August 04, 2010; revised January 07, 2011; accepted February 13, 2011. Date of publication April 05, 2011; date of current version April 06, 2012. This work was supported by UMC Inc., National Chip Implementation Center and National Science Council under Grant NSC95-2220-E-009-021.

The authors are with the Department of Electronics Engineering and Institute of Electronics, National Chiao Tung University, Hsinchu City 30050, Taiwan (e-mail: jackiewei.ee95g@nctu.edu.tw; jerryjou@iic94g.nctu.edu.tw; tschang@iic94g.nctu.edu.tw; shengj@iic94g.nctu.edu.tw).

Digital Object Identifier 10.1109/TVLSI.2011.2125805

TABLE I
COMPARISONS OF COMPLEXITY FOR FFT AND FHT

	FFT (real data) Split radix [16][19]		FHT [16][20][21]	
N	MUL.	ADD.	MUL.	ADD.
2	0	2	0	2
4	0	6	0	8
8	2	20	2	22
16	10	60	10	62
32	34	164	34	166
64	98	420	98	422

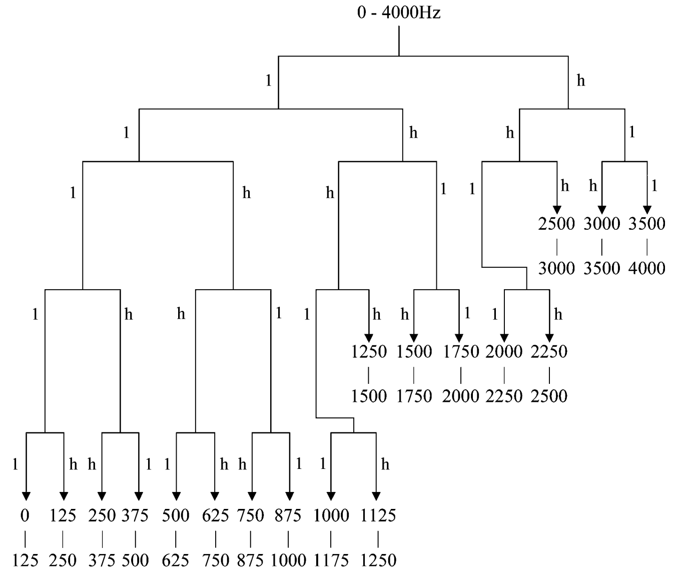


Fig. 1. P-DWPT with 18 subbands [23], l and h means low- and high-pass subfilter, respectively.

A. Related Works on Spectral Decomposition and Reconstruction

For spectral decomposition and reconstruction, perfect reconstruction methods are usually adopted, such as FFT [8] for uniform decomposition and reconstruction or DWT [9]–[15] for nonuniform decomposition and reconstruction.

FFT provides better frequency resolution. However, a long length FFT suffers from complicated computations, which increase power consumption. For simplifying FFT, many works have been proposed [16], such as Winograd [17], Cooley-Tukey [18], and split-radix [19]. Of those well-known algorithms, Winograd has the least complexity. However, corresponding architecture is very complicated and thus is not suitable for circuit implementation. Of the other two schemes with a regular structure, the split-radix has the lower computational complexity. In addition, the complexity of FFT is reduced if its input data are real numbers. Table I shows the complexity of the split radix FFT with real number input [16].

To reduce computational complexity, several modified transforms, such as FHT [20]–[22], whose equation is the same as that of FFT but without the j of the transfer kernel, have been proposed. Unlike FFT, FHT involves only real number computations and output spectra. Although FHT has almost the same complexity as FFT with real number inputs, it does not require magnitude and phase computations whose computational complexity cannot be neglected. Table I presents the complexity of FHT [16].

Since j is removed from the transfer kernel in FHT, the decomposition quality of FHT is degraded. For a single tone input, FFT yields high, concentrated energy at the associated frequency. In addition, the energy remains almost constant in every frame. In contrast, although the FHT has high and concentrated energy at the associated frequency, the energy in successive frame does not remain constant. Accordingly, using only the instantaneous magnitude will reduce the accuracy of the estimated signal power. To solve this problem, a moving average scheme is more suitable.

In contrast to FFT and FHT, DWT enables different basis choice for specific requirements to be met, such as computational complexity or time-frequency resolution. Moreover, DWT can provide a nonuniform decomposition. Exploiting these features, another modified transform, which first proposed by Coifman, discrete wavelet packet transform (DWPT) [15], is proposed to yield a specific resolution. The DWPT design that is based on human perception is called perceptual-DWPT (P-DWPT). It provides an efficient way to decompose and process acoustic signals. Hence, this transform is adopted as one of the decomposition and reconstruction tools herein. However, DWT has a poorer resolution in the frequency domain than FFT does, degrading the performance of VAD when frequency resolution is a concern in the denoise process. To solve this problem, for the proposed 4-zone-VAD in Section II-C, it detects voice based on the total energy of lower half band (0–2000 Hz), hence reducing the importance of frequency resolution.

Fig. 1 displays the filter bank realization of P-DWPT that was proposed by Lu [23], where l and h means low- and high-pass subfilter, respectively. The P-DWPT approximates the Bark scale and decomposes an input signal into 18 nonuniform subbands, based on the resolution of the human cochlear. Lu's filter bank requires 34 subfilters and the associated complexity depends on the selected wavelet basis. In the works [10] and [13], complex bases, such as 8-order Daubechies and 12-order Coiflet, are utilized. However, the complicated basis, such as the 8-order Daubechies, requires eight multiplications and seven additions in each subfilter, or a total of 272 multiplications and 238 additions in the decomposition. Moreover, this complexity is doubled when reconstruction is also considered .

To reduce the computational complexity, the Haar basis, whose lifting scheme subfilter [24] are shown in Fig. 2, can be adopted. Given this simple basis, this sub-filter requires one multiplication and one addition for each output, or 34 multiplications and 34 additions. Additionally, the number of multiplications can be further reduced by merging the multiplications of $\sqrt{2}$ at two successive subfilters and implementing them using one bit shift. Accordingly, the total number of multiplications in decomposition becomes only 13, or 26 if reconstruction is also considered.

Although using even levels of DWPT with Haar basis can eliminate multiplications, it is not a good solution for our algorithm. First, four levels of DWPT cannot provide enough number of subbands for low frequency band of perceptual decomposition. Second, although six levels of DWPT can offer

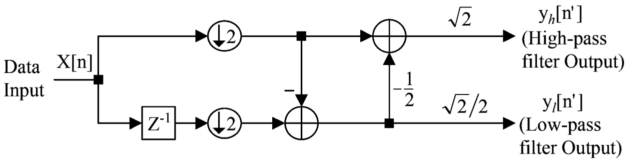


Fig. 2. Block diagram of Lifting subfilter of DWT with Haar basis.

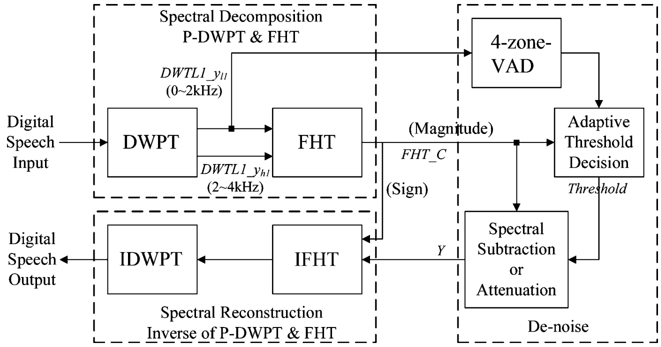


Fig. 3. Block diagram of the proposed noise cancellation algorithm.

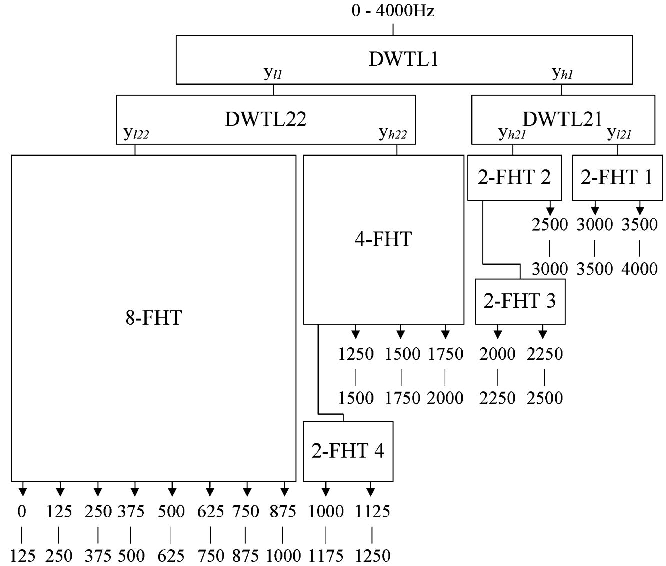


Fig. 4. Mixed P-DWPT&FHT decomposition, where DWT subfilter l and h are represented by DWT subfilters block depicted in Fig. 2.

enough subbands for low frequency band, it will cause 8 ms latency (64 samples with 8 kHz sampling rate), which is not acceptable for hearing aids. The maximum acceptable latency of hearing aids, according to previous research [25], is 15 ms. The 15 ms latency should be allocated to all components in hearing aids, such as analog front end, ADC, DAC, digital hearing prescription, noise reduction, and feedback cancellation. Therefore, five levels of decomposition (4 ms latency) is adopted for our decomposition structure.

B. Related Works on Denoise Process

The denoise process is composed of noise reduction and VAD. Noise reduction estimates the amount of noise and cancels it from the noisy signal. The cancellation performance can be improved by VAD, which detects the presence of human speech. However, traditional denoise processes such as Boll [8], Donoho [9], and Chen [10] use complicated computations, such as square roots or divisions for noise reduction or VAD, resulting in a high power consumption. Furthermore, these methods commonly suffer from consonant over-cancellation, which results in speech quality degradation.

C. Proposed Algorithm

Fig. 3 displays the block diagram of the proposed algorithm. The sampling rate is 8 kHz and the speech bandwidth is from 0 to 4 kHz. First, noisy speech is perceptually decomposed by a mixed decomposition consisting of perceptual-DWPT and FHT (P-DWPT&FHT) into various subbands and then processed using a 4-zone-VAD with an adaptive threshold for noise cancellation. Finally, the denoised speech is reconstructed using the inverse of P-DWPT&FHT.

The proposed mixed P-DWPT&FHT structure, shown in Fig. 4 combines the advantages of perceptual decomposition and low computational complexity. The 18 nonuniform subbands are like that shown in Fig. 1. The design guide line for mixing Haar basis DWPT and FHT is as follows.

1) The number of DWPT levels is even to avoid multiplications by $\sqrt{2}$.
2) The number of levels for FHT is one to preserve the decomposition property of FHT, or at most two to reduce property degradation and at the same time create nonuniform subbands.

To yield the minimum bandwidth of the subband, i.e., 125 Hz, 32 sampling data are used as a frame. The two levels of DWPT with Haar basis are adopted for the first uniform decomposition and one or two levels of FHT are used for further perceptual decomposition (including one 8-FHT, one 4-FHT and four 2-FHT). Table II shows the complexity of this mixed decomposition. The complexity of FFT and FHT are also listed for comparison. The results demonstrate that the total complexity of the proposed mixed P-DWPT&FHT, only two multiplications and 44 additions, is much lower than the complexities of the other three decompositions. Moreover, the performance of the five-levels P-DWPT and the proposed mixed P-DWPT&FHT method are similar, as shown in Section III. Therefore, we adopt the mixed P-DWPT&FHT for low power decomposition.

The denoise process consists of noise reduction (consisting of spectral subtraction or attenuation and adaptive threshold decision) and VAD. The proposed noise reduction method is based on Boll [8], noise-based [14] update flow and consonant protection for efficient denoising. This method, based on VAD decision, classifies a signal frame into four zones, which are strong voice, weak voice, quasi-noise and true noise, as displayed in Fig. 5. The strong voice and true noise zone are viewed as voice and noise region, respectively. The weak voice and quasi-noise zone are designed for transition region.

The concept of proposed denoise process is to apply weak denoising to the voice region to reduce noise while maintain speech structure and apply strong denoising to the noise only region to enhance noise reduction. Moreover, the transition region should be kept smooth from voice region to noise only region to avoid abrupt quality change. In which, the weak denoising

TABLE II

COMPARISONS OF COMPLEXITY FOR FFT (REAL DATA), FHT (DUMAMEL), P-DWPT (LU WITH HAAR BASIS) AND THE PROPOSED P-DWPT&FHT, NOTE THAT FFT HAS ADDITIONAL MAGNITUDE AND PHASE CALCULATION

			Complexity		
			Transform	***Magnitude***	***Phase***
FFT **(Real Data)**	**Split Radix**	MUL	34	32	0
		ADD	164	16	16
		Others	No	SQRT: 16	DIV: 16 LOG: 16
FHT	**Duhamel**	MUL	34	No	No
		ADD	166	No	No
P-DWPT	**Lu (Haar)**	MUL	13	No	No
		ADD	34	No	No
Proposed **P-DWPT&FHT**	**Lu (Haar)** **+ Duhamel**	MUL	8-DHT: 2	No	No
		ADD	44 (DWPT: 6, 8-DHT: 22, 4-DHT: 8, 4*2-DHT: 8)	No	No

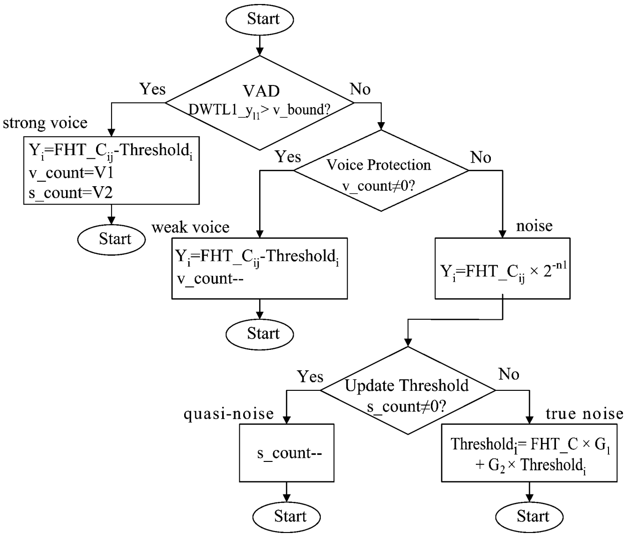


Fig. 5. Flow chart of denoise algorithm, where Y is denoise output and X is FHT output.

uses spectral subtraction, while the strong denoising uses spectral attenuation. For transition region, two counters (*v_count* and *s_count*) are utilized to control the transition period.

The denoise operation in Fig. 5 is described as follows. First, the presence or absence of speech is detected using a simple VAD, by determining whether $\mathrm{DWTL1\_}y_{l1}$ coefficient ($0 \sim 2$ kHz) exceed a preset voice bound *(v_bound)*. The *v_bound* being the VAD threshold is chosen by simulation according to noise power and type. It can be preset as a part of the scenarios for the hearing aids and can be selected by user.

In the strong voice zone, the estimated noise threshold is subtracted from the FHT output as follows:

$$Y_{ij} = \mathrm{FHT\_}C_{ij} - \mathrm{Threshold}_i \text{ for } i = 1 \text{ to } 18 \qquad (1)$$

where i is the subband index, j is time index at different subbands, Y is the denoise unit output, $\mathrm{FHT\_}C_{ij}$ is the FHT output and $\mathrm{Threshold}_i$ is the adaptive noise threshold. In the strong voice zone, a voice protection duration and quasi-noise control is set according to voice counter *(v_count)* and silence counter *(s_count)* to protect consonants and prevent abrupt threshold update, respectively, where $V1$ and $V2$ are preset constants for counter initialization.

For $\mathrm{DWTL1\_}y_{l1}$ that are smaller than *v_bound*, we will determine if the current speech frames are the ones following a strong voice zone. Such an input is identified as a weak voice zone and the spectral subtraction method, as shown in (1), is used for noise reduction. The weak voice zone remains only until the voice protection *(v_count)* expires. This mechanism protects the small consonant from over-reduction as a noise-only signal.

Outside the voice protection duration zone, the input is regarded as a noise-only signal and so is attenuated to remove noise by multiplications which can be replaced with $n1$-bit (in this case, $n1 = 5$) right shifts of the signal, as follows:

$$Y_i = \mathrm{FHT\_}C_{ij} \times 2^{-n1}. \qquad (2)$$

At the same time, the noise threshold should be gradually adapted to avoid abrupt variation. For this purpose, two mechanisms, quasi-noise zone and true noise zone, are used. In the quasi-noise zone, controlled by *s_count*, the threshold is kept constant. After the quasi-noise zone, the input frames without any speech detection sign will be regarded as noise only frames (true noise zone) and the noise threshold is updated using a moving average weighted scheme, owing to the FHT feature described in Section II, with only shifts and addition

$$\mathrm{Threshold}_i = \mathrm{FHT\_}C_{ij} \times G_1 + \mathrm{Threshold}_i \times G_2 \qquad (3)$$

where G_1 plus G_2 equals unity; in this paper, G_1 and G_2 are 0.25 and 0.75, respectively. In the above formulas, the constants are selected to be simple combinations of power-of-two digits to reduce computational complexity. The update scheme can be digitally programmed for different environments. Further complexity reduction can be attained using the speech-dependent skip scheme that is described in Section IV.

$V1$ and $V2$ (the initial value of *v_count* and *s_count*) can be set by the scenario control. $V1$ is selected based on environment and speech feature to protect small volume speech following large volume one. For high signal-to-noise ratio (SNR), $V1$ could be larger to protect speech from over-cancellation. For low SNR, it could be smaller since the low volume speech is strongly polluted by noise. $V2$ is designed to control the time we start to update $\mathrm{Threshold}_i$. The choice of $V2$ depends on the environment. For babble and factory noise which have larger variation compared to white noise, $V2$ should be set to be small value.

III. SIMULATION RESULTS

To evaluate the proposed algorithm, 20 3-s long fragments of Chinese speech with a 16-bits resolution are used for stimulus. All of these speeches are corrupted by three additive noises (white, speech babble and factory 1) from the NOISEX-92 database [26] and their segmental SNR [2] are evaluated. Fig. 6 plots the simulation results with a 5 dB SNR. Fig. 7 plots in detail the waveform associated with Fig. 6. These results reveal that the proposed method reduces the noise in the period of speech

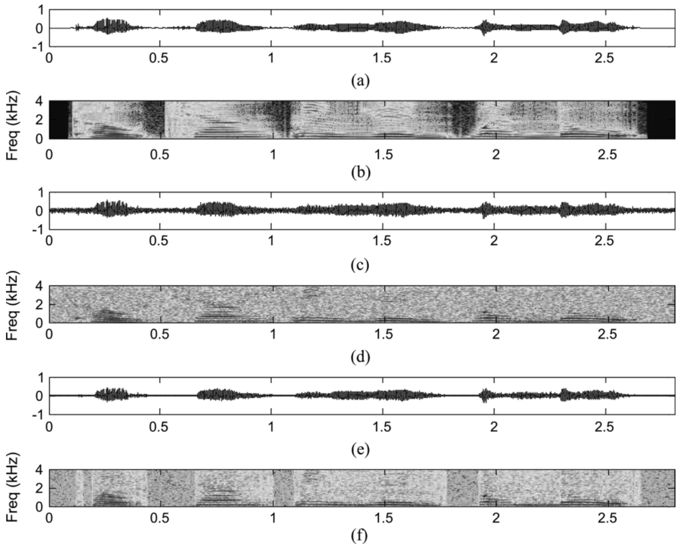


Fig. 6. Noise cancellation result: (a) original speech; (b) spectrogram of (a); (c) white noisy corrupted speech with 5 dB SNR; (d) spectrogram of (c); (e) result with proposed method; (f) spectrogram of (e).

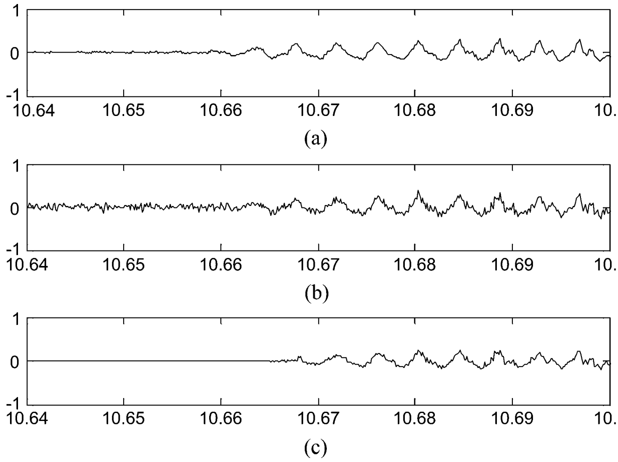


Fig. 7. Waveform detail from 10.64 to 10.70 s of Fig. 6: (a) original speech; (b) noisy speech with 5 dB SNR; (c) enhanced speech.

TABLE III
COMPARISONS OF SEGMENTAL SNR IMPROVEMENT IN DB FOR SPEECH CORRUPTED BY ADDITIVE (a) WHITE, (b) BABBLE, AND (c) FACTORY 1 NOISE

(A) White noise					
Input SNR	**-5dB**	**0dB**	**5dB**	**10dB**	**15dB**
specsub [12]	6.43	4.09	4.37	4.36	2.70
mband [27]	6.72	4.01	1.94	-1.34	-5.62
mmse [28]	9.00	7.08	4.72	2.01	-1.06
wiener_wt [32]	6.47	3.93	4.00	0.06	-2.53
mt_mask [29]	6.31	4.01	1.72	-0.68	-3.33
audnoise [30]	7.31	6.01	3.75	0.84	-2.33
pklt [31]	5.03	4.93	3.39	0.83	-2.48
P-DWPT + 4-zone-VAD de-noise	5.97	3.97	2.57	1.39	0.65
P-DWPT&DHT + 4-zone-VAD de-noise	5.95	3.99	2.61	1.43	0.66

(B) Babble noise					
Input SNR	**-5dB**	**0dB**	**5dB**	**10dB**	**15dB**
specsub [12]	5.76	3.55	2.95	2.23	1.21
mband [27]	3.76	2.78	1.17	-1.80	-1.07
mmse [28]	4.58	3.54	2.28	0.46	-1.91
wiener_wt [32]	5.71	2.53	0.17	-1.79	-3.74
mt_mask [29]	4.70	2.21	-0.67	-2.72	-4.47
audnoise [30]	4.58	2.69	1.10	-1.04	-3.42
pklt [31]	1.56	1.81	1.33	-0.62	-3.19
P-DWPT + 4-zone-VAD de-noise	4.06	2.79	1.28	0.41	0.00
P-DWPT&DHT + 4-zone-VAD de-noise	3.97	2.73	1.22	0.38	-0.01

(C) Factory 1 noise					
Input SNR	**-5dB**	**0dB**	**5dB**	**10dB**	**15dB**
specsub [12]	5.91	3.72	3.51	3.09	2.09
mband [27]	6.31	3.97	1.86	-1.49	-5.67
mmse [28]	6.22	5.00	3.51	1.38	-1.32
wiener_wt [32]	5.69	2.15	-0.28	-2.10	-3.87
mt_mask [29]	4.71	2.13	-1.10	-2.46	-4.48
audnoise [30]	5.68	2.77	1.67	-0.13	-2.87
pklt [31]	2.29	2.45	1.75	-0.26	-3.09
P-DWPT + 4-zone-VAD de-noise	4.97	3.27	1.99	0.99	0.38
P-DWPT&DHT + 4-zone-VAD de-noise	4.87	3.17	1.94	0.98	0.38

but maintains its formants and that the noise in the period of non-speech is also significantly reduced.

Table III shows the average segmental SNR improvement for various input SNR of the proposed algorithm and compares them with those of other FFT [12], [27]–[31] or wavelet-based [32] methods, provided by Loizou [2] for speech corrupted by additive white, babble and factory 1 noise. The results demonstrate that for all input SNR the proposed mixed P-DWPT&FHT exhibits similar performance to the P-DWPT when the proposed 4-zone-VAD denoise method is used.

A hearing loss person needs better SNR compared to a normal hearing person, if both of them want to understand the same amount of speech. According to previous research [33], the average SNR deficit of mild hearing loss people is about 4 dB. For a normal hearing person, a speech with SNR of 5 dB can be well recognized. According to Table III, in the case with white noise and a SNR above 5 dB, the proposed algorithm can further enhance the SNR. For 10 dB or especially 15 dB, the proposed algorithm outperforms most of methods, because the voice protection and quasi-noise strategy prevents over-cancellation of speech and makes noise estimation more stable to avoid abrupt quality change, respectively. In a case of low SNR, such as -5 and 0 dB, the proposed algorithm can improve SNR up to 5.95 and 3.99 dB which are comparable with the others, except mmse. The mmse can theoretically provide very good performance with very high complexity, hence being difficult to be utilized for hearing aids.

The speech quality is further evaluated by Itakura-Saito (IS) distance[2] and is shown in Table IV. The IS is a measurement of the spectrum difference between an original and a distorted speech. Small IS means more similarity between two speech. The results indicate that both of the proposed mixed and P-DWPT methods improve the quality of noisy speech for all input SNR. Furthermore, both methods represent stable improvements over others and provide a similar speech quality to that obtained using complicated methods such as mband and mmse, because of their associated voice protection and quasi-noise strategy.

When input speech is corrupted by babble or factory 1 noise, all algorithms perform worse than they do in the white noise environment. In these two non-white noise environments, babble

TABLE IV
COMPARISONS OF IS DISTANCE FOR SPEECH CORRUPTED BY ADDITIVE (a) WHITE, (b) BABBLE, AND (c) FACTORY 1 NOISE

(A) White noise					
Input SNR	**-5dB**	**0dB**	**5dB**	**10dB**	**15dB**
Noisy Speech	6.42	5.41	4.45	3.57	2.79
specsub	12.16	11.21	6.48	4.60	5.32
mband	4.47	3.77	3.09	2.63	2.32
mmse	4.16	3.40	2.85	2.49	2.34
wiener_wt	74.64	70.75	69.46	68.26	65.00
mt_mask	12.33	26.16	41.72	45.26	48.83
audnoise	79.30	73.62	70.83	68.54	64.19
pklt	24.26	47.94	53.84	55.88	56.25
P-DWPT + 4-zone-VAD de-noise	3.95	3.93	3.08	2.51	1.81
P-DWPT&DHT + 4-zone-VAD de-noise	4.03	3.94	3.08	2.51	1.81

(B) Babble noise					
Input SNR	**-5dB**	**0dB**	**5dB**	**10dB**	**15dB**
Noisy Speech	4.52	3.63	2.85	2.18	1.61
specsub	11.09	10.25	7.07	5.48	5.68
mband	4.01	3.28	2.57	2.17	2.08
mmse	3.98	3.36	2.89	2.61	2.38
wiener_wt	40.72	40.69	41.40	43.13	44.14
mt_mask	28.66	35.46	43.74	44.40	42.84
audnoise	47.66	43.61	41.67	41.86	42.38
pklt	25.96	35.19	39.19	41.43	45.78
P-DWPT + 4-zone-VAD de-noise	3.26	2.73	1.78	1.15	0.76
P-DWPT&DHT + 4-zone-VAD de-noise	3.29	2.58	1.78	1.15	0.76

(C) Factory 1 noise					
Input SNR	**-5dB**	**0dB**	**5dB**	**10dB**	**15dB**
Noisy Speech	5.30	4.33	3.45	2.69	2.03
specsub	9.98	9.50	6.60	5.19	4.86
mband	3.76	3.32	2.82	2.33	2.01
mmse	3.92	3.32	2.85	2.57	2.39
wiener_wt	61.13	61.35	58.92	57.11	56.26
mt_mask	29.95	37.92	46.10	49.68	49.19
audnoise	66.14	62.96	57.56	54.02	53.96
pklt	29.04	40.39	46.50	47.78	50.84
P-DWPT + 4-zone-VAD de-noise	3.35	2.71	2.06	1.47	1.12
P-DWPT&DHT + 4-zone-VAD de-noise	3.28	2.72	2.07	1.48	1.12

noise is associated with the poorest performance because it occupies the same bandwidth as the speech of interest. Additionally, several algorithms, such as specsub, mband and mmse, cause quality degradation at high SNR. This effect may result from over-cancellation that is caused by the relatively abrupt change of non-white noise.

Although our proposed P-DWPT&FHT and denoise process with 4-zone-VAD have much less complexity, the performance is comparable with other methods except mmse, as show in Tables III and IV. This is because the P-DWPT&FHT performs similar decomposition performance compared to pure P-DWPT with much less complexity. In addition, the 4-zone VAD and denoise provides smooth denoising with small volume speech protection and stable noise estimation. Therefore, the proposed method can provide similar improvement in low SNR and better performance in high SNR.

In summary, the proposed algorithm performs noise reduction which is comparable to other complicated algorithms, but with much less complexity. Furthermore, the proposed algorithm provides a better speech quality, owing to its voice protection and quasi-noise strategy, especially at high SNR.

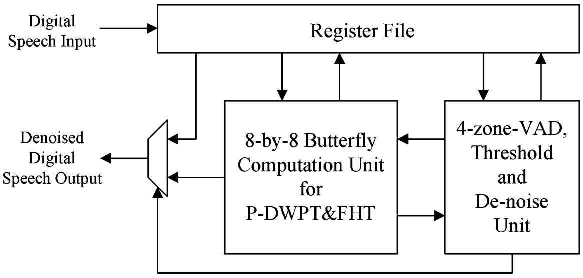

Digital Speech Input
Register File
Denoised Digital Speech Output
8-by-8 Butterfly Computation Unit for P-DWPT&FHT
4-zone-VAD, Threshold and De-noise Unit

Fig. 8. Implementation architecture of the proposed algorithm.

IV. DESIGN OF ARCHITECTURE ASSOCIATED WITH THE PROPOSED ALGORITHM

Fig. 8 displays the proposed architecture, which comprises an 8-by-8 butterfly-based computation unit for the mixed P-DWPT&FHT, a 4-zone-VAD, threshold and denoise unit for denoise process and a register file for data storage. For P-DWPT&FHT, DWPT, FHT, and their inverse methods are folded into a configurable 8-by-8 butterfly computation unit with on-time scheduling. The 4-zone-VAD, threshold and denoise unit also contains a simple finite state machine that is controlled by VAD to control the operations of the circuit. The register file stores inputs and data that are generated by the two blocks. Moreover, a skip scheme is designed for reconstruction to further reduce power consumption. The whole operation is consistent with the algorithm that was proposed in Section II-C. Given an 8 kHz sampling clock for a 4 kHz speech bandwidth, each 32 sampled speech data are treated as a frame. In the 32 clock cycles, frequency decomposition, 4-zone-VAD, threshold and denoise and reconstruction of the noise reduced speech shall be carried out with acceptable extra latency.

To establish a low-power architecture for implementing the proposed noise reduction algorithm, very regular and yet flexible computational units are required. The regular architecture results in low interconnect complexity so the wire loading and routing area are small. Also, the delay paths are balanced so that timing constraints are easily achieved and unnecessary glitches are avoided. The flexible architecture enables different computations to be easily mapped into these computational units and the scheduling is very simple, such that the clock rate and the fetch/store of temporal data is as low as possible. These blocks are described in detail below, based on the above design guide lines.

A. *8-by-8 Configurable Butterfly Computation Unit With On-Time Scheduling*

FHT algorithms that exploit partial sum sharing are also suitable for low power implementation. In this paper, Hou's algorithm [22] is adopted, because of its regular architecture and effectiveness for sharing resources among FHTs of various sizes. Fig. 9 plots the signal flow graph of Hou's 2-FHT, 4-FHT, and 8-FHT, which are composed of 1, 4, and 12 2-by-2 butterflies, respectively.

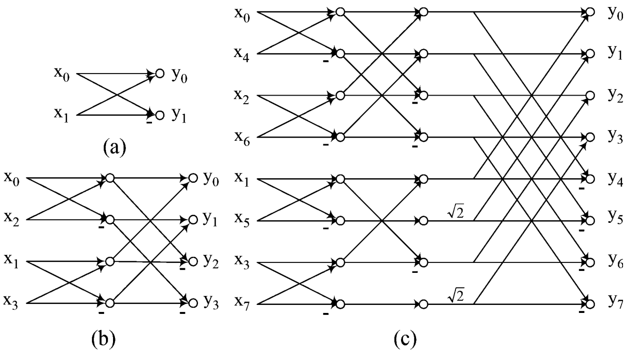


Fig. 9. Signal flow graph of Hou's [22]: (a) 2-FHT, (b) 4-FHT, and (c) 8-FHT, where ○ means addition.

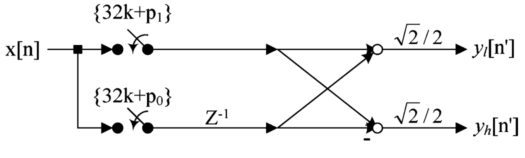


Fig. 10. Signal flow graph of butterfly for Haar subfilter, where $p0 = \{0, 2, 4, \ldots, 30\}$ and $p1 = \{1, 3, 5, \ldots, 31\}$ are the cycles for switch closing in each frame and outputs are available at $\{32k + p1\}$.

For DWT implementation, the low-pass filter output $y_l[n']$ and high-pass filter output $y_h[n']$ in Fig. 2 can be rewritten as (4) and a more symmetric 2-by-2 butterfly structure is obtained, as shown in Fig. 10

$$y_l\,[n'] = \frac{\sqrt{2}}{2}(x[2n+1] - x[2n])$$
$$y_l\,[n'] = \frac{\sqrt{2}}{2}(x[2n+1] - x[2n]) \tag{4}$$

where the sampling frequency of $y_l[n']$ and $y_h[n']$ is one half of $x[n]$.

Although the butterfly DWT subfilter has the same computational complexity as the original lifting one, it has balanced delay paths (which reduce glitch for low power consumption) and more importantly, it has the same butterfly structure as that of FHT, thereby simplifying computation unit design.

For the architecture design of spectral decomposition and reconstruction unit, the hardware area and operation frequency shall be chosen to make power consumption very low. For different design tradeoffs (from unfolded to fully folded), the hardware cost for datapath is linearly scaled according to the folding factor while the overhead of control cost is relatively small according to our experience. However, direct folding implies linearly increased clock rate by the folding factor according to design practices.

Our major design goal is to satisfy the latency constraint with lowest possible frequency and hardware cost. For this design, the lowest possible frequency for folded design is the data rate, 8 kHz, as that in non-folded design. For latency constraint, according to Section II, we allocate at most 4.5 ms (4 ms is caused by frame length and 0.5 ms is resulted from four cycles of data processing) for latency of noise reduction. Accordingly, the remaining question is to design a efficient architecture to complete all computations within 4.5 ms latency (or 36 clock cycles).

Fig. 11 shows the original computation schedule of the decomposition and reconstruction of each frame with 32 cycles latency, where DWT and IDWT are scheduled by pyramid scheduling. The superscription k means the frame number. According to the original schedule, the first 31 cycles (from $\{32k + 0\}$ to $\{32k + 30\}$) can be utilized for data input and some computations that do not have data dependency. However, the $\{32k+31\}$ cycle has the maximum computational load, consisting of one DWT and IDWT, two 8-FHTs, two 4-FHTs, and eight 2-FHTs. Hence, there are 42 2-by-2 butterflies should be finished. In addition, owing to the four extra latency, the computations of DWPT and IDWPT for the first four cycles of the next frame (from $\{32(k + 1) + 0\}$ to $\{32(k + 1) + 3\}$) should be finished as well. Therefore, overall 48 2-by-2 butterflies have to be completed within 4 cycles. It turns out that in each clock we should finish 12 2-by-2 butterflies. This implies a partial folded 8-by-8 butterfly architecture which has 12 2-by-2 butterflies.

Cycle	***Decomposition***	***Reconstruction***
32k+0	DWT^{k}L1(1)	IDWT^{k-1}L1(1)
32k+1	DWT^{k}L21(1)	IDWT^{k-1}L22(1)
32k+2	DWT^{k}L1(2)	IDWT^{k-1}L1(2)
32k+3	DWT^{k}L22(1)	IDWT^{k-1}L21(2)
32k+4	DWT^{k}L1(3)	IDWT^{k-1}L1(3)
32k+5	DWT^{k}L21(2)	IDWT^{k-1}L22(2)
32k+6	DWT^{k}L1(4)	IDWT^{k-1}L1(4)
32k+7	DWT^{k}L22(2), 2-FHT1^{k}(1), 2-FHT2^{k}(1)	IDWT^{k-1}L21(3), 2-IFHT1^{0}(2), 2-IFHT2^{0}(2)
32k+8	DWT^{k}L1(5)	IDWT^{k-1}L1(5)
32k+9	DWT^{k}L21(3)	IDWT^{k-1}L22(3)
32k+10	DWT^{k}L1(6)	IDWT^{k-1}L1(6)
32k+11	DWT^{k}L22(3)	IDWT^{k-1}L21(4)
32k+12	DWT^{k}L1(7)	IDWT^{k-1}L1(7)
32k+13	DWT^{k}L21(4)	IDWT^{k-1}L22(4)
32k+14	DWT^{k}L1(8)	IDWT^{k-1}L1(8)
32k+15	DWT^{k}L22(4), 2-FHT1^{k}(2), 2-FHT2^{k}(2), 2-FHT3^{k}(1), 4-FHT^{k}(1)	IDWT^{k-1}L21(5), 2-IFHT1^{k-1}(3), 2-IFHT2^{k-1}(3), 2-IFHT3^{k-1}(2), 4-IFHT^{k-1}(2)
32k+16	DWT^{k}L1(9)	IDWT^{k-1}L1(9)
32k+17	DWT^{k}L21(5)	IDWT^{k-1}L22(5)
32k+18	DWT^{k}L1(10)	IDWT^{k-1}L1(10)
32k+19	DWT^{k}L22(5)	IDWT^{k-1}L21(6)
32k+20	DWT^{k}L1(11)	IDWT^{k-1}L1(11)
32k+21	DWT^{k}L21(6)	IDWT^{k-1}L22(6)
32k+22	DWT^{k}L1(12)	IDWT^{k-1}L1(12)
32k+23	DWT^{k}L22(6), 2-FHT1^{k}(3), 2-FHT2^{k}(3)	IDWT^{k-1}L21(7), 2-IFHT1^{k-1}(4), 2-IFHT2^{k-1}(4)
32k+24	DWT^{k}L1(13)	IDWT^{k-1}L1(13)
32k+25	DWT^{k}L21(7)	IDWT^{k-1}L22(7)
32k+26	DWT^{k}L1(14)	IDWT^{k-1}L1(14)
32k+27	DWT^{k}L22(7)	IDWT^{k-1}L21(8)
32k+28	DWT^{k}L1(15)	IDWT^{k-1}L1(15)
32k+29	DWT^{k}L21(8)	IDWT^{k-1}L22(8)
32k+30	DWT^{k}L1(16)	IDWT^{k-1}L1(16)
32k+31	DWT^{k}L22(8), 2-FHT1^{k}(4), 2-FHT2^{k}(4), 2-FHT3^{k}(2), 4-FHT^{k}(1), 2-FHT4^{k}(2), 8-FHT^{k}(1)	IDWT^{k}L21(1), 2-IFHT1^{k}(1), 2-IFHT2^{k}(1), 2-IFHT3^{k}(1), 2-IFHT4^{k}(1), 4-IFHT1^{k}(1), 8-IFHT1^{k}(1)

Fig. 11. Computation schedule of mixed decomposition and reconstruction.

Table V shows the comparison for fully, partial folded and unfolded architectures that all meet 36 cycles latency. The extra storage means the registers requirement in addition to input data storages. For the hardware utilization rate, the numerator is the total number of 2-by-2 butterflies that should be completed within one frame. The denominator is the product of frame length, folding factor, and number of 2-by-2 butterflies.

The fully folded architecture has the minimum area for arithmetic; however, it needs more extra storages (eight 16-bits registers) for temporary storage. Additionally, it has complicated

TABLE V
COMPARISONS OF DIFFERENT ARCHITECTURES THAT ALL MEET 36 CYCLES LATENCY

	Fully Folded	*Partial Folded*	*Unfolded*
Arithmetic Cost (2-by-2 Butterfly)	One butterfly	12 butterflies	23 butterflies
Extra Storages	4+8	4	4
Control Cost	Very Complicated	Middle	Simple
Clock Rate	8KHz*12	8KHz	8KHz
Utilization Rate	33.9% (130/384)	33.9% (130/384)	17.7% (130/736)

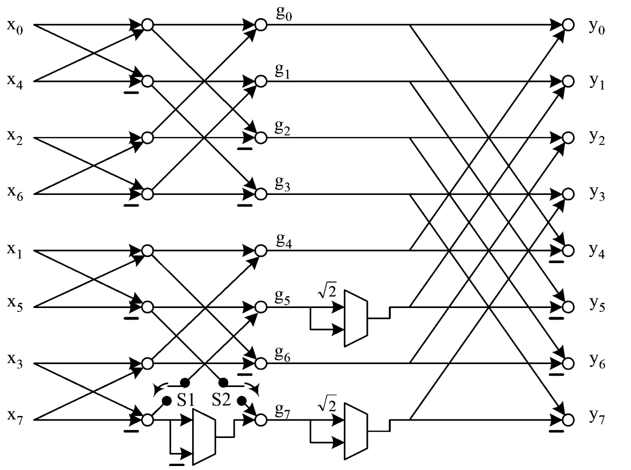
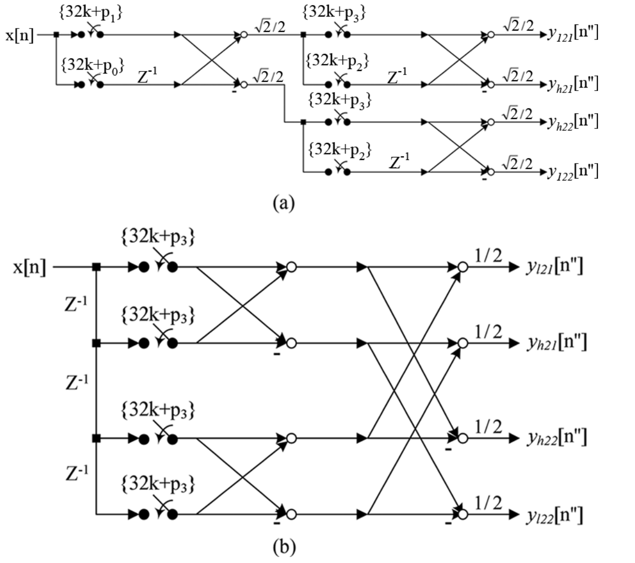


Fig. 12. Proposed computation unit for spectrum decomposition and reconstruction, where each butterfly is the same as Fig. 10 and switch S_1 and S_2 are used for butterfly enabling.

control and clock rate is increased to 96 kHz due to so many computations in the required latency. For power consumption, the power of computation unit may be similar to partial and unfolded architecture with operand isolation scheme. However, the 12 times clock rate will increase the power of storages, control, and clock tree. Furthermore, it can be a limitation to voltage scaling.

On the contrary, the partial folded architecture has about 1/2 hardware cost compared to unfolded one. The cost of control is similar to unfolded architecture (according to our synthesis report), but will be much simpler compared to fully folded since folding FHT is very complicated. The clock rate is the same as data rate (8 kHz), yet the utilization rate is equal to the one of fully folded. Moreover, the larger area of unfolded design will cause more leakage power if advanced process is adopted. Therefore, the 8-by-8 butterfly is a good choice for the tradeoff between power and area with 36 cycles latency.

Hence, a computation unit based on this 8-by-8 butterfly structure with some control signals and multiplexers, as displayed in Fig. 12, is proposed. The computation unit can be configured for all types of DWPT, FHT, and their inverses, given smart construction and an evenly distributed loading in one frame. Note that by using two 4-by-4 butterflies, owing to data dependency, the extra latency will be increased by 8 cycles. Furthermore, by using one 4-by-4 butterfly, the extra latency will be greatly increased by 20 cycles. In addition, the fetch/store operations is dramatically increased as well.



(a)

(b)

Fig. 13. Signal flow graph of (a) paralleled DWT and (b) PBDWT, where $p0$ and $p1$ are the same as Fig. 10, $p2 = \{1, 5, 9, 13, 17, 21, 25, 29\}$, $p3 = \{3, 7, 11, 15, 19, 23, 27, 31\}$.

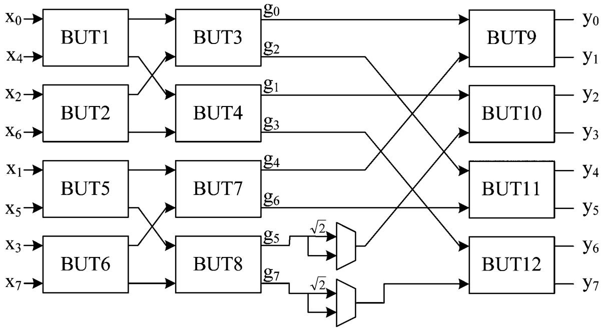
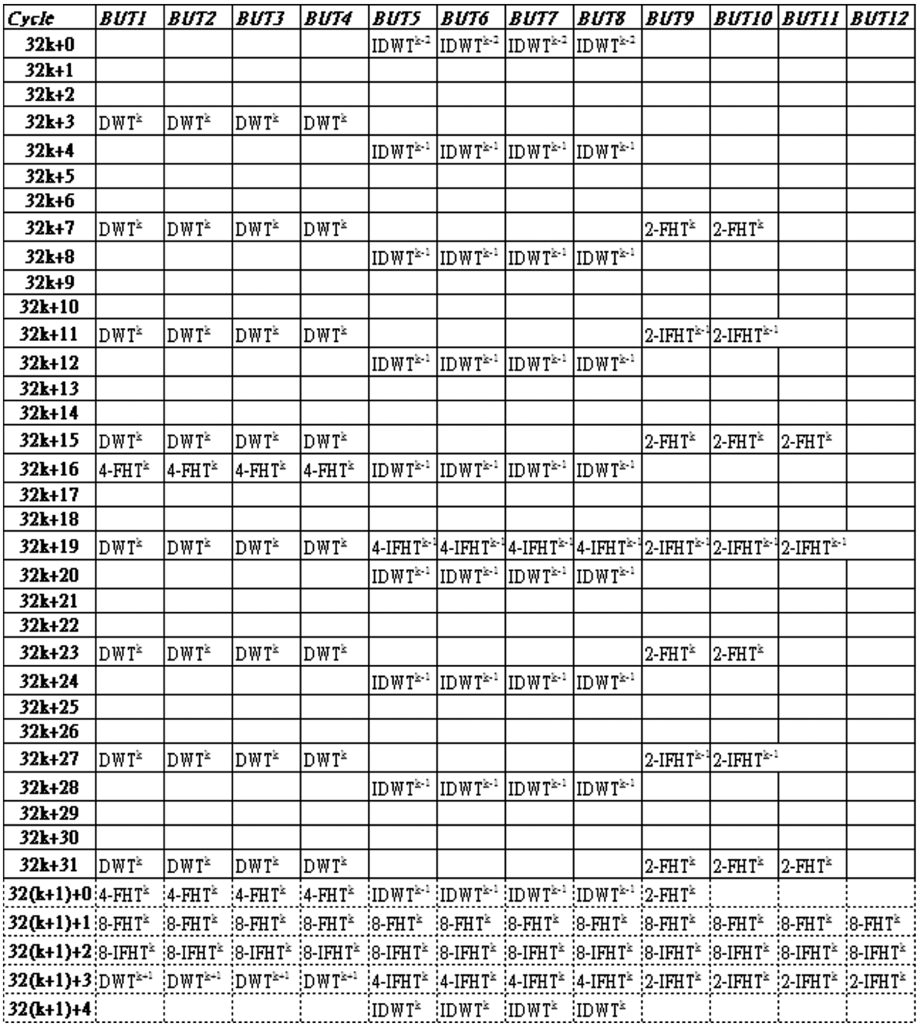


Fig. 14. Another representation of Fig. 11 with butterfly number, where BUT means a two-input two-output butterfly.

Given the computation unit shown in Fig. 12, the mapping of the FHT signal flow of Figs. 9–12 is quite straightforward. For DWPT, the power consumption can be reduced by the parallel scheduling of the corresponding two-stage pyramid DWPT. The parallel architecture is displayed in Fig. 13(a), where the outputs, whose sampling frequency is down sampled by four, are available at the $\{32k + p_3\}$ cycles in each frame. This architecture enables the second DWT stage to use the outputs of the first stage as soon as they are generated, thereby reducing the amount of fetch/store of data. In addition, the two storages in the second-stage input in Fig. 13(a) can be further moved to the first stage by the retiming method, yielding a 4-by-4 butterfly architecture, called parallel butterfly-based DWT (PBDWT) and shown in Fig. 13(b), where the outputs are available at the $\{32k + p_3\}$ cycles in each frame. Notably, the PBDWT is identical to the 4-FHT that is shown in Fig. 9(b). Accordingly, the DWPT and FHT can be mapped into the same butterfly architecture, simplifying the computational unit and the schedule design.

The final design and schedule are derived as follows. Given four cycles extra latency, the 48 2-by-2 butterflies resulted from two PBDWT, two 8-FHTs, two 4-FHTs, and eight 2-FHTs in cycle $\{32k+31\}$ of Fig. 11 can be mapped to the 8-by-8 butterflies, as shown in Fig. 12, with some configurability of input and



Cycle	BUT1	BUT2	BUT3	BUT4	BUT5	BUT6	BUT7	BUT8	BUT9	BUT10	BUT11	BUT12
32k+0					IDWT^{k-2}	IDWT^{k-2}	IDWT^{k-2}	IDWT^{k-2}				
32k+1												
32k+2												
32k+3	DWT^{k}	DWT^{k}	DWT^{k}	DWT^{k}								
32k+4					IDWT^{k-1}	IDWT^{k-1}	IDWT^{k-1}	IDWT^{k-1}				
32k+5												
32k+6												
32k+7	DWT^{k}	DWT^{k}	DWT^{k}	DWT^{k}					2-FHT^{k}	2-FHT^{k}		
32k+8					IDWT^{k-1}	IDWT^{k-1}	IDWT^{k-1}	IDWT^{k-1}				
32k+9												
32k+10												
32k+11	DWT^{k}	DWT^{k}	DWT^{k}	DWT^{k}					2-IFHT^{k-1}	2-IFHT^{k-1}		
32k+12					IDWT^{k-1}	IDWT^{k-1}	IDWT^{k-1}	IDWT^{k-1}				
32k+13												
32k+14												
32k+15	DWT^{k}	DWT^{k}	DWT^{k}	DWT^{k}					2-FHT^{k}	2-FHT^{k}	2-FHT^{k}	
32k+16	4-FHT^{k}	4-FHT^{k}	4-FHT^{k}	4-FHT^{k}	IDWT^{k-1}	IDWT^{k-1}	IDWT^{k-1}	IDWT^{k-1}				
32k+17												
32k+18												
32k+19	DWT^{k}	DWT^{k}	DWT^{k}	DWT^{k}	4-IFHT^{k-1}	4-IFHT^{k-1}	4-IFHT^{k-1}	4-IFHT^{k-1}	2-IFHT^{k-1}	2-IFHT^{k-1}	2-IFHT^{k-1}	
32k+20					IDWT^{k-1}	IDWT^{k-1}	IDWT^{k-1}	IDWT^{k-1}				
32k+21												
32k+22												
32k+23	DWT^{k}	DWT^{k}	DWT^{k}	DWT^{k}					2-FHT^{k}	2-FHT^{k}		
32k+24					IDWT^{k-1}	IDWT^{k-1}	IDWT^{k-1}	IDWT^{k-1}				
32k+25												
32k+26												
32k+27	DWT^{k}	DWT^{k}	DWT^{k}	DWT^{k}					2-IFHT^{k-1}	2-IFHT^{k-1}		
32k+28					IDWT^{k-1}	IDWT^{k-1}	IDWT^{k-1}	IDWT^{k-1}				
32k+29												
32k+30												
32k+31	DWT^{k}	DWT^{k}	DWT^{k}	DWT^{k}					2-FHT^{k}	2-FHT^{k}	2-FHT^{k}	
32(k+1)+0	4-FHT^{k}	4-FHT^{k}	4-FHT^{k}	4-FHT^{k}	IDWT^{k-1}	IDWT^{k-1}	IDWT^{k-1}	IDWT^{k-1}	2-FHT^{k}			
32(k+1)+1	8-FHT^{k}	8-FHT^{k}	8-FHT^{k}	8-FHT^{k}	8-FHT^{k}	8-FHT^{k}	8-FHT^{k}	8-FHT^{k}	8-FHT^{k}	8-FHT^{k}	8-FHT^{k}	8-FHT^{k}
32(k+1)+2	8-IFHT^{k}	8-IFHT^{k}	8-IFHT^{k}	8-IFHT^{k}	8-IFHT^{k}	8-IFHT^{k}	8-IFHT^{k}	8-IFHT^{k}	8-IFHT^{k}	8-IFHT^{k}	8-IFHT^{k}	8-IFHT^{k}
32(k+1)+3	DWT^{k+1}	DWT^{k+1}	DWT^{k+1}	DWT^{k+1}	4-IFHT^{k}	4-IFHT^{k}	4-IFHT^{k}	4-IFHT^{k}	2-IFHT^{k}	2-IFHT^{k}	2-IFHT^{k}	2-IFHT^{k}
32(k+1)+4					IDWT^{k}	IDWT^{k}	IDWT^{k}	IDWT^{k}				

Fig. 15. On-time schedule of the proposed computation unit.

path selection. Fig. 14 displays another representation of Fig. 12 which all 2-by-2 butterfly units are replaced by blocks with assigned number. Thus, a mapping can be carried out by firstly decomposing all PBDWT and FHT into combinations of 2-by-2 butterfly units and mapping these computations into the 8-by-8 butterfly. If two computation paths on the 8-by-8 butterfly can be scheduled without conflict, then they can be merged into a single 8-by-8 butterfly and so the computations can be performed in a single cycle. Fig. 15 presents the resulting on-time schedule. An example of mapping can be presented by the $\{32k+7\}$ cycle in Fig. 11. In that cycle, the decomposition involves the computing of one PBDWT and two 2-FHT computations. Mapping the PBDWT at BUT1 to BUT4 as in Fig. 14, the two 2-FHT mapped at BUT9 and BUT10 can directly get their inputs (the outputs of PBDWT) from g_0 to g_3 without any path conflict.

Further power savings can be achieved by exploiting the cascaded butterfly. Fig. 16 illustrates one example in which the second FHT stage can begin immediately as soon as all inputs have been generated. Six fetch/store operations are saved per frame or 32 inputs. In addition, 32 fetch/store operations are saved if the IFHTs are begun immediately as soon as the denoise process has been completed.

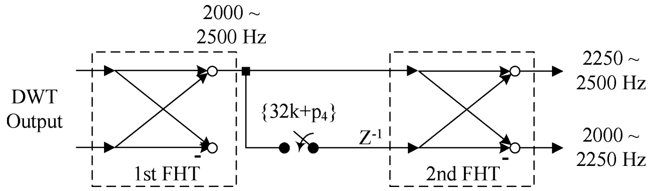


Fig. 16. Example of cascaded FHT, where $p_4 = \{7, 23\}$.

B. Skip Scheme for Reconstruction

Power consumption is reduced by avoiding unnecessary reconstruction when speech frames are in the quasi and true noise zone. The output signal y that is reconstructed from attenuated denoise results can be written as

$$y = \mathrm{IDWT}\left(\mathrm{IFHT}\left(2^{-n1} \times X\right)\right) = 2^{-n1}x \qquad (5)$$

where X is $\mathrm{FHT}(\mathrm{DWT}(x))$. The constant 2^{-n1} can be moved out of the parentheses because of the linear property of these transforms. Hence, all of the forward and backward transforms cancel each other. Accordingly, inverse transforms of the quasi

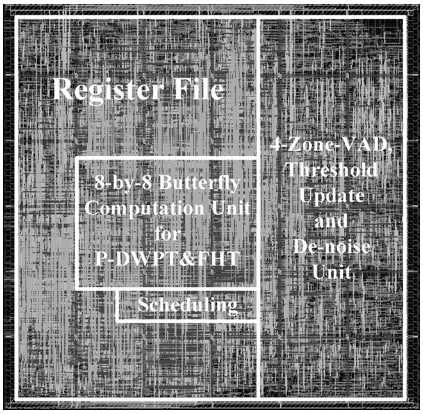

Register File
8-by-8 Butterfly Computation Unit for P-DWPT&FHT
Scheduling
4-Zone-VAD, Threshold Update and De-noise Unit

Fig. 17. Chip layout of the proposed design.

TABLE VI
IMPLEMENTATION SUMMARY OF CHIP HARDWARE AND POWER CONSUMPTION

CHIP PARAMETERS	***RESULT***
Process	0.18um 1.8V CMOS
Gate Count	29016
Area	770um * 750um
Clock Rate	8KHz
Data Rate	8KHz
Total Power Consumption	2.58uW @ 1.8V; 0.65uW@ 1.0 V

and true noise zone are avoided and power consumption is greatly reduced.

V. Results of VLSI Implementation

The proposed architecture is implemented using hardware description language Verilog and synthesized with a 0.18-μm CMOS standard cell library. The design target is operation with a 1.0 V power supply. To reduce dynamic power further, clock gating and operand isolation are utilized. The circuit layout is implemented by auto-placement and routing tool. Fig. 17 displays the layout of the design.

According to the area report, the gate count of control circuit for on-time scheduling is only about 1138 (4% of the total gate count). The circuit only consists of a 5-bits counter and several constant comparators to generate control signals. Since the gate count of one 8-by-8 butterfly is 3878, the hardware sharing do reduce the total area.

To verify the function correctness and the power consumption, circuit extracted from layout is simulated using HSPICE. According to the results, the circuit works correctly down to 0.9 V. The power consumption is 2.58 and 0.65 μW at 1.8 and 1.0 V, respectively. Table VI summarizes the implementation of the chip design. It costs only 29 016 equivalent gates and therefore can be used as a dedicated ASIC module, or as a dedicated accelerator or attached processor for DSP processor system, to relax the computational requirement of noise cancellation and to reduce the power consumed of the hearing aid chip. Accordingly, the algorithm, architecture and circuit optimization cause the proposed noise reduction module for CIC hearing aids to consume only sub μW power. Comparison with other single microphone designs or the design of spectral subtraction algorithms is difficult to perform since none has been dedicated to the noise reduction function of VLSI implementation.

VI. Conclusion

This paper presents a low-power method of noise reduction for CIC hearing aids. For spectral decomposition and reconstruction, novel mixed P-DWPT&FHT are proposed with 18 sub-bands, nonuniform, perceptual distribution, and one-sixth the complexity of the pure P-DWPT method. A simple yet efficient denoise and 4-zone-VAD with consonant protection and an adaptive noise threshold are proposed to reduce noise while provide high speech quality with simple hardware comprising shifters and adders. A partial folded 8-by-8 butterfly-based computation unit with a on-time scheduling is proposed to reduce the switching rate and hardware cost, hence reducing the power consumption with acceptable latency. Additionally, a skip scheme, enabled by VAD, is proposed to further reduce the power consumed by reconstruction. Simulations reveal that the proposed noise reduction method reduces noise with low complexity and performance similar to that of the highly complicated algorithms. Moreover, circuit implementation that is based on a 0.18-μm CMOS standard cell library demonstrates that the proposed architecture consumes only 0.65 μm of power at 1.0 V.

References

[1] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1985.

[2] P. C. Loizou, *Speech Enhancement, Theory and Practice*. Boca Raton, FL: CRC Press, 2007.

[3] F. Carbognani, F. Burgin, L. Henzen, H. Koch, H. Magdassian, C. Pedretti, H. Kaeslin, and N. Felber, "A 0.67-mm^2 45-μW DSP VLSI implementation of an adaptive directional microphone for hearing aids," in *Proc. Eur. Conf. Circuit Theory Des. III*, 2005, pp. 141–144.

[4] C. H. Kim, H. Soeleman, and K. Roy, "Ultra-low-power DLMS adaptive filter for hearing aid applications," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 11, no. 6, pp. 1058–1067, Dec. 2003.

[5] P. Mosch, G. V. Oerle, S. Menzl, N. Rougnon-Glasson, K. V. Nieuwenhove, and M. Wezelenburg, "A 660-μW 50-Mops 1-V DSP for a hearing aid chip set," *IEEE J. Solid-State Circuits*, vol. 35, no. 11, pp. 1705–1712, Nov. 2000.

[6] O. Parker, J. Sparso, N. Isager, L. S. Nielsen, and J. Melanson, "A heterogeneous multiprocessor architecture for low-power audio signal processing applications," in *Proc. IEEE Comput. Soc. Workshop VLSI*, 2001, pp. 47–53.

[7] T. Stetzler, N. Magotra, P. Gelabert, P. Kasthuri, and S. Bangalore, "Low power real-time programmable DSP development platform for digital hearing aids," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1999, pp. 2339–2342.

[8] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 27, no. 2, pp. 113–120, Apr. 1979.

[9] D. L. Donoho, "De-noising by soft-thresholding," *IEEE Trans. Inf. Theory*, vol. 41, no. 3, pp. 613–627, May 1995.

[10] S. H. Chen and J. F. Wang, "Speech enhancement using perceptual wavelet packet decomposition and teager energy operator," *J. VLSI Signal Process. Syst.*, vol. 36, no. 2-3, pp. 125–139, Feb.-Mar. 2004.

[11] D. L. Donoho and I. M. Johnstone, "Ideal spatial adaptation via wavelet shrinkage," *Biometrika*, vol. 81, no. 3, pp. 425–455, 1994.

[12] M. Berouti, M. Schwartz, and J. Makhoul, "Enhancement of speech corrupted by acoustic noise," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 1979, pp. 208–211.

[13] Q. Fu and E. Wan, "Perceptual wavelet adaptive denoising of speech," in *Proc. 8th Eur. Conf. Speech Commun. Technol.*, 2003, pp. 577–580.

[14] S. Ayat, M. T. Manzuri-Shalmani, and R. Dianat, "An improved wavelet-based speech enhancement by using speech signal features," *Comput. Elect. Eng.*, vol. 32, no. 6, pp. 411–425, Aug. 2006.

[15] S. Mallat, *A Wavelet Tour of Signal Processing, The Sparse Way*, 3rd ed. San Diego, CA: Academic Press, 2008.

[16] G. Bi and Y. H. Zeng, *Transforms and Fast Algorithms for Signal Analysis and Representations*. Boston, MA: Birkhauser, 2003.

[17] S. Winograd, "On computing the discrete Fourier transform," *Proc. Nat. Acad. Sci. USA, Math.*, vol. 73, no. 4, pp. 1005–1006, Apr. 1976.

[18] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Math. Comput.*, vol. 19, no. 90, pp. 297–301, Apr. 1965.

[19] P. Duhamel, "Implementation of split-radix FFT algorithms for complex, real and real-symmetric data," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 34, no. 2, pp. 285–295, Apr. 1986.

[20] P. Duhamel and M. Vetterli, "Improved Fourier and Hartley transform algorithm: Application to cyclic convolution of real data," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-35, no. 6, pp. 818–824, Jun. 1987.

[21] H. V. Sorensen, D. L. Jones, C. S. Burrus, and M. T. Heideman, "On computing discrete hartlet transform," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-33, no. 4, pp. 1231–1238, Oct. 1985.

[22] H. Hou, "The fast hartley transform algorithm," *IEEE Trans. Comput.*, vol. C-36, no. 2, pp. 147–156, Feb. 1987.

[23] C. T. Lu and H. C. Wang, "Enhancement of single channel speech based on masking property and wavelet transform," *Speech Commun.*, vol. 41, no. 2-3, pp. 409–427, Oct. 2003.

[24] W. Sweldens, "The lifting scheme: A new philosophy in biorthogonal wavelet constructions," in *Proc. SPIE Wavelet Appl. Signal Image Process. III*, 1995, pp. 68–79.

[25] M. A. Stone and B. C. J. Moore, "Tolerable hearing aid delays. II. estimation of limits imposed during speech production," *Ear Hear.*, vol. 23, no. 4, pp. 325–338, Aug. 2002.

[26] A. Varga, H. J. M. Steenneken, M. Tomlinson, and D. Jones, "NOISEX-92," 1992 [Online]. Available: http://spib.rice.edu/spib/select_noise.html

[27] S. Kamath and P. Loizou, "A multi-band spectral subtraction method for enhancing speech corrupted by colored noise," in *Proc. IEEE Int. Conf. Acout. Speech Signal Process.*, 2002, pp. 2–5.

[28] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean square error short time spectral amplitude estimator," *IEEE Trans. Acoust., Speech Signal Process.*, vol. 32, no. 6, pp. 1109–1121, Dec. 1984.

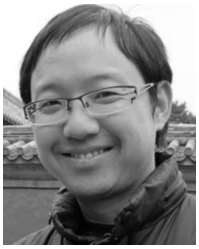
[29] Y. Hu and P. Loizou, "Incorporating a psychoacoustical model in frequency domain speech enhancement," *IEEE Signal Process. Lett.*, vol. 11, no. 2, pp. 270–273, Feb. 2004.

[30] D. E. Tsoukalas, J. N. Mourjopoulos, and G. Kokkinakis, "Speech enhancement based on audible noise suppression," *IEEE Trans. Speech Audio Process.*, vol. 5, no. 6, pp. 497–514, Nov. 1997.

[31] F. Jabloun and B. Champagne, "Incorporating the human hearing properties in the signal subspace approach for speech enhancement," *IEEE Trans. Speech Audio Process.*, vol. 11, no. 6, pp. 700–708, Nov. 2003.

[32] Y. Hu and P. Loizou, "Speech enhancement based on wavelet thresholding the multitaper spectrum," *IEEE Trans. Speech Audio Process.*, vol. 12, no. 1, pp. 59–67, Jan. 2004.

[33] H. Dillon, *Hearing Aids*. New York: Boomerang Press, 2001.



Cheng-Wen Wei received the B.S. and M.S. degree in electrical engineering from the Yuan-Ze University, Taoyuan, Taiwan, in 1998 and 2000, respectively. He is currently pursuing the Ph.D. degree in electronic engineering from National Chiao-Tung University (NCTU), Hsinchu, Taiwan, since 2006.

From 2000 to 2006, he was an Engineer and worked on Delta Sigma data convertor, speech signal processing and VLSI design, with the Product Development Division/Digital Circuit Design Department (DCD), Elan Microelectronics Corporation (EMC), Hsinchu, Taiwan. Since 2006, he has been a consultant with the DCD, EMC. His research interests include digital signal processing, speech processing, data conversion, and low power VLSI design.

Sheng-Jie Su received the M.S. degrees in electronic engineering from National Chiao-Tung University (NCTU), Hsinchu, Taiwan, in 2008.

He is currently an Engineer with the Power Department, Asus Corporation Inc., Taipei, Taiwan. His current research interests include power system design and system integration.

Tian-Sheuan Chang (S'93–M'06–SM'07) received the B.S., M.S., and Ph.D. degrees in electronic engineering from National Chiao-Tung University (NCTU), Hsinchu, Taiwan, in 1993, 1995, and 1999, respectively.

He is currently an Associate Professor with the Department of Electronics Engineering, NCTU. From 2000 to 2004, he was a Deputy Manager with Global Unichip Corporation, Hsinchu, Taiwan. His current research interests include (silicon) intellectual property (IP) and system-on-a-chip design, VLSI signal processing, and computer architecture.

Shyh-Jye Jou (S'86–M'90–SM'97) received the B. S. degree in electrical engineering from National Chen Kung University, Tainan, Taiwan, in 1982 and M.S. and Ph.D. degrees in electronics from National Chiao Tung University, Hsinchu, Taiwan, in 1984 and 1988, respectively.

He joined the Electrical Engineering Department, National Central University, Chung-Li, Taiwan, from 1990 to 2004, and became a Professor in 1997. Since 2004, he has been Professor of the Electronics Engineering Department, National Chiao Tung University, and became the Chairman from 2006 to 2009. He was a visiting research Professor with the Coordinated Science Laboratory, University of Illinois, Urbana-Champaign, during 1993–1994 and 2010 academic years. In the summer of 2001, he was a visiting research consultant with the Communication Circuits and Systems Research Laboratory of Agere Systems.

Prof. Jou has served on the technical program committees in CICC, A-SSCC, ICCD, ISCAS, ASP-DAC, VLSI-DAT, and other international conferences. He has published over 100 IEEE journal and conference papers. His research interests include design and analysis of high speed, low power mixed-signal integrated circuits, communication, and bio-electronics integrated circuits and systems.