Hardware design for Blind Source Separation using a Fast Time-Frequency Mask Technique

Tsung-Han Tsai (han@ee.ncu.edu.tw)
National Central University https://orcid.org/0000-0001-7524-0621
Pei-Yun Liu
Universidad Nacional de la Patagonia San Juan Bosco Biblioteca Central Dr Eduardo Musacchio
Yu-He Chiou
National Central University

Research

Keywords: Blind separation, Time frequency mask, Convolutive BSS, Reduction of DOA variance, VLSI Design

Posted Date: February 28th, 2020

DOI: https://doi.org/10.21203/rs.3.rs-15403/v1

Version of Record: A version of this preprint was published at Integration on August 1st, 2021. See the published version at https://doi.org/10.1016/j.vlsi.2021.07.001.

PAPER

Hardware design for Blind Source Separation using a Fast Time-Frequency Mask Technique

Tsung-Han Tsai, Pei-Yun Liu, Yu-He Chiou
han@ee.ncu.ed.tw, daisyliu@dsp.ee.ncu.edu.tw, a879156@dsp.ee.ncu.edu.tw

SUMMARY: In this paper, we propose a fast time-frequency mask technique for blind source separation in order to separate a mixture of two input sounds in single signal automatically. Mostly previous methods utilize a linear sensor array, and therefore they cannot separate symmetrically positioned sources. To overcome such problems, we first define two features which are normalized level-ratio and phase-difference. Next, we use our method to decrease Direction of Arrival (DOA), this can reduce the variance of features so that it can reduces iterations of k-means. Finally, according to the clustered features, a mask is generated. Our method does not require any prior information or parameter estimation and we have made a real demonstration system. We use Signal to Distortion Ratio (SDR) and Signal to Interference Ratio (SIR) to compare our method. Then we present hardware design. Hardware design uses TSMC 90-nm CMOS process. As a cost-effective result, it consumes about 120K gates and executes with frequency of 10MHz. The power consumption is only 2.92 mW with low power design considerations.

Key words: *Blind separation, Time frequency mask, Convolutive BSS, Reduction of DOA variance, VLSI Design.*

1.Introduction

Blind source separation (BSS) is a technique to estimate individual source components from their mixtures at multiple sensors. In general, this is a difficult problem due to several complicated factors. One reason is that the signal reaching a microphone has several noises along with data signal such as room reverberations and echoes. The other reason is that in some simple mixing models, each recording consists of a sum of differently weighted source signals. Furthermore, in many real-world applications, such as in acoustics, the mixing process is more complex. In such systems, the mixtures are weighted and delayed, where each source contributes to the sum with multiple delays corresponding to the multiple paths by which an acoustic signal propagates to a microphone. Such filtered sums of different sources are called convolutive mixtures. In these situations, the sources are the desired signals, yet only the recordings of the mixed sources are available while the mixing process is unknown. Thus, BSS is a challenging problem in real room environments.

Applications of the BSS technique for speech include hands-free teleconference systems [1] and automatic conference minute generators. More specifically, one promising application is a car navigation system [2]. In general, we can distinguish two cases depending on the number of N sources and the number of M sensors [3]; (i) $N>M$ is the underdetermined BSS and (ii) $N\leq M$ is the (over-) determined BSS. Since over-determined BSS ($N<M$) can be reduced to determined BSS ($N=M$), we refer to both as determined BSS. Most approaches deal with determined BSS, but in reality, BSS is often underdetermined.

Two approaches have been widely studied and employed to solve the BSS problem: one is based on statistics such as Independent Component Analysis (ICA) [4], [5] and the other approach relies on the sparseness of source signals [6]. As shown in Table 1, the method used for solution of statistically independent latent variables is called the independent components analysis. ICA works well even in a reverberant condition in the (over-) determined condition. It takes the advantage of high precision, but the robustness can be low. ICA can perform in several domains including the time-domain BSS [7], frequency-domain BSS [8], and the hybrid time- and frequency- domain BSS [9].

Table 1. Relationship between BSS.

Approach	Method	Characteristics	Algorithm
Sparseness	MAP estimation	Low computation	[6],[10],[11]
	Binary mask	High robustness, low presicion	[12],[13],[14], [15],[19],[20]
Statistics	ICA	High preicsion, Low robustness	[4],[5],[7], [8],[9],[15]

On the other hand, the sparseness-based approaches are attractive because they can cope with the underdetermined problem. It takes the advantage of high robustness but the precision could be low. The sparseness-based methods can be categorized into two main approaches. One method is based on Maximum a Posteriori (MAP) estimation [10] [11] where the sources are estimated after mixing matrix estimation. And the other method is based on binary mask where we can extract each signal with time-frequency binary masks [12] [13]. The MAP approach investigated the consequence of dealing with complex numbers as a result of the time-frequency domain approach. Although the combinatorial solution with at least N-M zeros is not theoretically justified for complex numbers, its performance quality is comparable to or even better than that of the Second-order Cone Programming (SOCP) solution. In addition, the combinatorial solution has the advantage that it is faster for solving underdetermined BSS problems with low input/output dimensions [10].

Binary mask approach is based on Direction of Arrival (DOA) estimation for sources and the inter frequency correlation. It has the advantages of high robustness and being implemented in real time [14]. As shown in Table 1, robustness and preciseness are the two key features to evaluate the performance of BSS [15].

In this paper, an effective algorithm for the BSS of speech sources is proposed. It combines time frequency mask with decreasing DOA variance in feature extraction. The algorithm has the advantage of low complexity and saving more storage. In addition, it does not degrade the quality of the separated signals. The VLSI architecture of BSS are introduced below. In [16], matrix whitening algorithm is proposed for easy hardware structure. And [17] talks about convolutive blind source separation using TSMC90nm process to implement infomax filtering module and scaling factor computation module used for BSS. And in [18], a low-power ICA architecture with

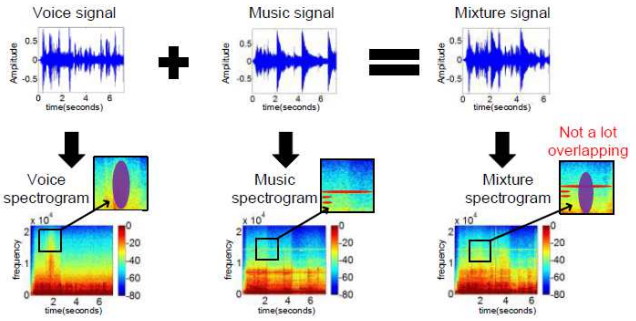

Fig. 1 Time frequency spectrum of mixed input signal.

outer-product learning rules is designed for separating method.

This paper is organized as follows: In ection 2, we introduce the background of BSS based on sparseness. In Section 3, the proposed structure is introduced. In Section 4, we present hardware design. Section 5 presents the experimental results of software algorithm and hardware testing, and sections 6 is conclusion.

2. Background on Sparseness-based BSS

2.1 Sparse component analysis

Switching to the time-frequency domain has the additional advantage of making it easier to exploit the time-frequency sparseness of speech sources [19]. Sparseness of a signal means the places where only a few instances have a value significantly different from zero. The higher sparseness in the time-frequency domain can be explained by the harmonic structure of speech signals. During voiced speech, the energy of a speech signal is concentrated around multiples of the speaker's fundamental frequency. Ideally, the frequency bands in between do not carry any energy. This means that in the time-frequency domain, only a few frequency bins have high values at each time instance, while most frequency bins have a value close to zero. This is a sparse signal by definition. Together with the frequency sparseness and the speaker dependency altogether leads to less overlap, which is also known as 'disjoint' or W-disjoint orthogonality [20], in the time frequency domain. Using a sparse signal representation is very important in order to ensure good separation performance since the separation is built on the assumption of sparse source signals.

An example is presented in Fig. 1. We assume there is a mixture of music and voice signals. Then, we transform these signals to frequency domain by Short Time Fourier Transform (STFT) and obtain the spectrogram. In the spectrogram, third dimension indicates the amplitude of a particular frequency at a specific time represented with color. In the mixture, it is obvious that its spectrogram retains the characteristics of voice and music. Fig. 1 also illustrates the sparseness of source signals. We can observe that the percentage of low energy bins i.e. are blue, are more than high energy bin which are red. It proves that the frequency bands in between do not carry any energy in ideal conditions. And we can distinguish that the red part stands for a high intensity of music and purple represents a high intensity of voice. In the spectrogram of the mixture, their time-frequency bins do not overlap significantly.

2.2 Sparseness-based approaches

The sparseness-based approaches can be divided into two main categories. One method is based on MAP estimation [10] [11], where the sources are estimated after mixing matrix estimation, while the other extracts each signal with time-frequency binary masks [12]-[13]. The former method includes mixing matrix estimation and L1-norm minimization in the frequency domain (i.e., for complex numbers), both of which still present difficulties [10]. The latter binary mask approach has the advantage of fast implementation.

A basic MAP estimation design is used which is explained in [10]. It used two-stage approach consisting of Blind Mixing Model Recovery (BMMR) and Blind Source Recovery (BSR). A hierarchical cluster is used to estimate the mixing matrix in the BMMR step. Eventually the system separates the signals in the BSR step. Then, the inverse STFT is applied to obtain time-domain signals. Among the most important advantages of the described hierarchical clustering algorithm, there is a fact that it works directly on the sample data in any vector space of arbitrary dimensions. The only requirement is the definition of a distance measure for the considered vector space. Therefore, it can easily be applied to complex valued data that occurs in frequency-domain convolutive BSS. No initial values for the mixing vectors are required. This means, in particular, that if the assumption of clusters with high densities around the mixing vectors is true, then the algorithm converges to those clusters.

In the binary mask approach, the signals are sufficiently sparse. Therefore, we can assume that at most one source is dominant at each time–frequency slot. If this assumption holds, a histogram of the level and frequency of normalized phase differences [21] between two sensor observations has N clusters. Because an individual cluster in the histogram corresponds to an individual source, each signal can be separated by selecting the observation signal at time–frequency points in each cluster with a binary mask.

3. Proposed Binary Mask Approach on BSS

3.1 Overall System Description

Suppose that N sources are convolutively mixed and observed at M sensors.

$$x_p(t) = \sum_{}^{N} \sum h_{pq}(l) s_q(t-1), \qquad p = 1, \dots, M \quad (1)$$

Where $h_{pq}(l)$ represents the impulse response from source q to sensor p, $x_p(t)$ represents the sample from microphone, and $s_q(t-l)$ represents the source signal. We assume that N and M are known, and that the sensor spacing is small enough to avoid the spatial aliasing problem. The goal is to obtain separated signals that are estimations of sources solely from M observations.

Five steps of binary mask approach are shown in Fig. 2 which are discussed as following:

Step 1) Signal transformation to the time-frequency domain: the signals sampled at frequency f_s are converted into frequency-domain by STFT. As following equation:

$$X(t,f) = \int_{-\infty}^{\infty} w(t-\tau) x(\tau) e^{-j2\pi f \tau} d\tau \qquad (2)$$

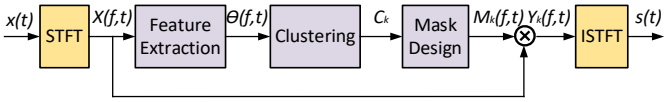

Fig. 2 Functional block of binary mask approach.

STFT adds a shifted window function *w(t)* to Fourier Transform to divide the signal into several blocks then every block is transformed by Fourier Transform.

Step 2) Feature extraction: If the sources are sufficiently sparse, separation can be realized by gathering the time-frequency points. To estimate such time-frequency points, some features are calculated by using the frequency-domain observation signals. Furthermore, feature is a vector that consists of certain geometric features. Generally, previous methods utilized the level ratio and/or phase difference between observations as their features.

Step 3) Clustering: The clustering criterion is to minimize the total sum of the Euclidean Distances (ED) between cluster members and their centroids. Features are grouped into *N* clusters where *N* is the number of possible sources. Here we use *k*-means algorithm. Therefore, the clustering procedure will be automated and simplified. In this paper, *N* is set to 2.

Step 4) Next, the separated signals are estimated based on the clustering results. We can design a time–frequency domain binary mask that extracts the time–frequency points of each cluster. If some of the features belong to one group, we set $M(f, t) = 1$, otherwise 0, and the mask is generated. Then, we multiply the binary mask to the mixture spectrogram. In the same time, it separates the signal.

$$M_k(f,t) = \begin{cases} 1, \Theta(f,t) \in C_k \\ 0, otherwise \end{cases} \qquad (3)$$

$$y_k(f,t) = M_k(f,t)x_p(f,t) \qquad (4)$$

Step 5) Separated signal reconstruction: At the end of the flow in Fig. 2, the output *y(t)* is obtained by employing an ISTFT and the overlap-and-add method.

3.2 Proposed Modified Feature Extraction

Traditional feature extraction is usually applied with *k*-means algorithm. It assumes that the distributions of isotropic variance, the level of ratios and the phase differences should have similar variances. Here we propose a modified feature extraction which introduces the reduction of DOA variance during feature extraction. After applying this method, iterations of *k*-means clustering can be decreased. Generally, features can be summarized as:

$$\Theta(\mathrm{f,t}) = \left[\left[\frac{|x_2(f,t)|}{|x_1(f,t)|}, arg\,\frac{|x_2(f,t)|}{|x_1(f,t)|}\right]\right]^T \qquad (5)$$

Such features represent geometric information on sources and sensors if the sources are sufficiently sparse. Let us assume that the mixing process is expressed as:

$$h_{jk} \approx \lambda_{jk} exp\left[-j2f\tau_{jk}\right] \qquad (6)$$

If the sources are sparse, then the feature vector becomes:

$$\Theta(\mathrm{f,t}) = \left[\frac{\lambda_{2k}}{\lambda_{1k}}, -2\pi f(\tau_{2k} - \tau_{1k})\right]^T \qquad (7)$$

To avoid frequency dependency in the phase difference, some authors have employed a frequency normalization that involves dividing the phase difference by $2\pi f$ or $2\pi fc^{-1}d$ where *c* is the propagation velocity and *d* is the sensor spacing. The latter gives the DOA of sources if the sensor spacing *d* is given correctly. If we do not use such frequency normalization, then we have to solve the permutation problem among frequencies after clustering the features. Moreover, frequency normalization makes it possible to apply the method to short data without significant performance degradation. Thus, we have to find a feature that leads to accurate centroid estimates blindly.

Some method applies level ratio for evaluation [20], which is listed in equation (8). Another method which concerns the normalization of level ratio is listed in equation (9). These two types are manipulated for two sources and two microphones case, where *d* is the space between microphones and *c* is the sound velocity. Arakia *et al*. [22] mentioned that although these equations are similar, the feature of *Type_B* can achieve better performance than *Type_A*. It is found that when the feature is normalized, the level ratios as seen in feature of *Type_B* can prevent such outliers. Another reason is that the phase term of *Type_A* feature is too small, and his is more important and more fatal. For multivariate clustering with the *k*-means algorithm, the level ratios and phase differences should have similar variances. This is because the *k*-means assumes distributions of isotropic variance. However, the phase term of *Type_A* feature is far smaller than the level ratio. The poor performance of *Type_A* feature results is due to lack of balance between the level ratio and phase difference terms. With the feature of *Type_B*, where the phase is divided by $2\pi fc\text{-}1d$, the phase difference becomes larger, and achieves good performance with the *k*- means algorithm.

$$Type\_A:\ \Theta(\mathrm{f,t}) = \left[\left[\frac{|x_2(f,t)|}{|x_1(f,t)|}, \frac{1}{2\pi f} arg\,\frac{|x_2(f,t)|}{|x_1(f,t)|}\right]\right]^T \qquad (8)$$

$$Type\_B:\ \Theta(\mathrm{f,t}) = \left[\left[\frac{|x_2(f,t)|}{A(f,t)}, \frac{1}{2\pi fc^{-1}d} arg\,\frac{|x_2(f,t)|}{|x_1(f,t)|}\right]\right]^T \qquad (9)$$

We choose *Type_B* and the feature extraction formula are described as following:

$$\Theta^L(\mathrm{f,t}) = \left[\frac{|x_2(f,t)|}{A(f,t)}\right] \qquad (10)$$

$$\Theta^P(\mathrm{f,t}) = \left[\frac{1}{\alpha f} arg\left[\frac{|x_2(f,t)|}{|x_1(f,t)|}\right]\right] \qquad (11)$$

Where $\alpha = 2\pi c^{-1}d$

$$\mathrm{A(f,t)} = \sqrt{|x_1(f,t)|^2 + |x_2(f,t)|^2} \qquad (12)$$

Thus, the feature is expressed as:

$$\Theta(\mathrm{f,t}) = \Theta^{L}(f,t)exp[j\Theta^{P}(f,t)] \qquad (13)$$

Then, normalize the feature by:

$$\widehat{\Theta}(\mathrm{f,t}) \leftarrow \Theta(f,t)/\Theta|(f,t)| \qquad (14)$$

3.3 Reduction of DOA variance

Even though we normalized features in order to make level ratio and phase difference have similar variance, the variance of the phase difference is still very small. Thus, we propose the reduction of DOA variance [11]. The equation is shown as following:

$$\hat{\theta} = \sqrt{\varepsilon} * \theta + \left(1 - \sqrt{\varepsilon}\right)u \qquad (15)$$

The variance σ^2 can be adjusted by ε as shown below:

$$\hat{\sigma}^2 = \varepsilon\sigma^2 \qquad (16)$$

Where μ is the mean of phase difference, θ is the phase difference, and σ is set to 0.5. The advantage of this method is that it will decrease the iterations of *k*-means.

The procedure of *k*-means is first setting initial centroids and then assigning the data to the nearest centroid. During this step, it will engender clusters. Afterwards, the centroid of a cluster will be recalculated. Then, we can repeat these steps until it converges. Since *k*-means uses the distance information to assign data to the nearest centroid and the variance of phase term is still larger than the level ration term, we can reduce the variance of phase term. Thus, the total sum of distance decreases and the average iteration of clustering is reduced.

4. Hardware design

In hardware design, the first module is combined by STFT and ISTFT in one module, second module i.e. cluster module is combined by k-means and binary mask. Therefore, we have three main modules; STFT/ISTFT, feature extraction and cluster and buffer memory. First source signal is converted to frequency domain using FFT-based processing. The features generated by the feature extraction module are used in cluster module. The cluster module is executed for several iterations, which means that feature will go as input into the cluster module for several times. Therefore, buffer memory is needed. By using ISTFT, the signal is transformed back to time domain. The architecture is shown in Fig. 3. The detailed explanation of these architectures is provided below.

4.1 Architecture of Fast Fourier Transform

The transformation between time domain and frequency domain of the discrete Fourier transform (DFT) can be expressed as equation (19) and equation (20),

$$X[k] = \sum_{n=0}^{N-1} x[n]W_N^{kn}, k = 0,1,\dots,N-1 \qquad (19)$$

$$X[n] = \frac{1}{N}\sum_{n=0}^{N-1} X[k]W_N^{-kn}, k = 0,1,\dots,N-1 \qquad (20)$$

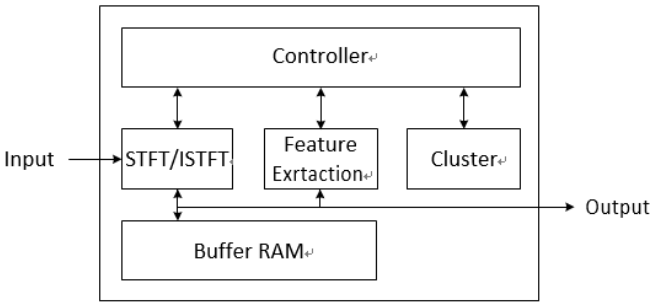

Fig. 3 Architecture diagram of BSS.

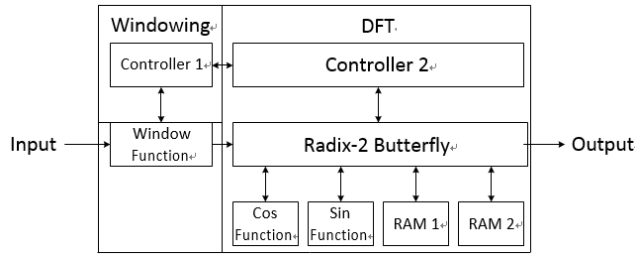

Fig. 4 Architecture diagram of STFT.

where x_n and X_k are the signals in time domain and frequency domain respectively. In this work, *N* is set as 512. When the period is power of 2, we can use radix-2 Fast Fourier transform (FFT) algorithn to compute discrete Fourier transform [23], [24]. We separate the points of X_k into even and odd point. After serveral computation, the equations of the DFT can be simplifed as equation (21) and (22):

$$X[2r] = \sum_{n=0}^{\frac{N}{2}-1}\left(x[n] + x\left[n+\frac{N}{2}\right]\right)W_{\frac{N}{2}}^{nr} \qquad (21)$$

$$X[2r+1] = \sum_{n=0}^{\frac{N}{2}-1}\left(x[n] - x\left[n+\frac{N}{2}\right]\right)W_{\frac{N}{2}}^{nr} \qquad (22)$$

where $W_N^{-kn} = e^{-j2\pi kn/N}$ is called twiddle factor. Equation (21) can be regarded as the sum of the first *N/2* points and the last *N/2* points, and then discrete Fourier transform is computed for *N/2* points. Similarly, equation (22) can also be regarded as the subtraction of the first *N/2* points and the last *N/2* points, and then computation of discreate Fourier transform for *N/2* points.

In this study, we use 512-points radix-2 memory based FFT architecture as shown in Fig. 4. In STFT module, we have one buterfly unit, coefficient generater and single-port SRAM. The size of SRAM is 512 words. The control decides the forward/inverse FFT opteration and generates the address which is required by RAM and coefficient generater for access. The cofficient generater includes window function and *sin(cos)* function that output the windowing and twiddle factor for further butterfly processing. Then the butterfly processing perform the complex arithmetic operations for the data of RAM and the twiddle factor. The RAM stores the input data as well as the temporary computed data and also play the role of the cache. The outputs of the FFT/IFFT results are also stored in the RAM for access.

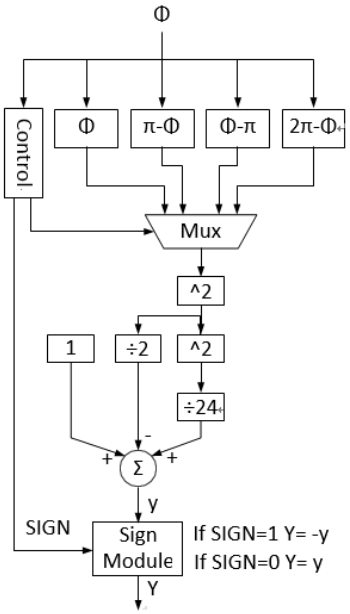

Fig. 5 Architecture diagram of fourth-order Maclaurin series.

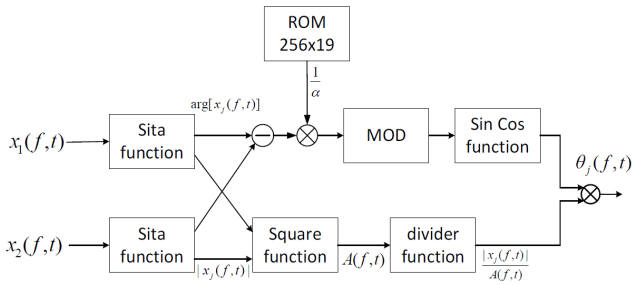

Fig. 6 Architecture diagram of feature extraction.

Maclaurin series expansions and the symmetric and periodic properties of the trigonometric functions are used to simplify the transform operation. In order to reduce the hardware complexity, our FFT processors have employed the Maclaurin series architecture to perform windowing and twiddle factor operations. It also has potential advantage of low switching activity for low-power operations.

Here, we use fourth-order Maclaurin series for the approximation of $\cos(\phi)$, which is given as follows:

$$\cos(\emptyset) = 1 - \frac{\emptyset^2}{2!} + \frac{\emptyset^4}{4!} - \frac{\emptyset^6}{6!} + \cdots \approx 1 - \frac{\emptyset^2}{2!} + \frac{\emptyset^4}{4!} \quad (23)$$

A block diagram of the proposed architecture based on the fourth-order Maclaurin series [25] is shown in Fig. 5. Input signal ϕ is simultaneously computed for four arguments (ϕ, $\pi-\phi$, $\phi-\pi$, $2\pi-\phi$) so that one of the four arguments can fall within the range of 0 to $\pi/2$. The controller module produces the 'select' signal that selects corresponding argument to be selected as the output of the multiplexer. Also the controller module outputs the 'SIGN' signal, which is dependent on the value of input ϕ. Absolute result of STFT will be changed to the negative value if the 'SIGN' is high.

4.2 Architecture of Feature Extraction

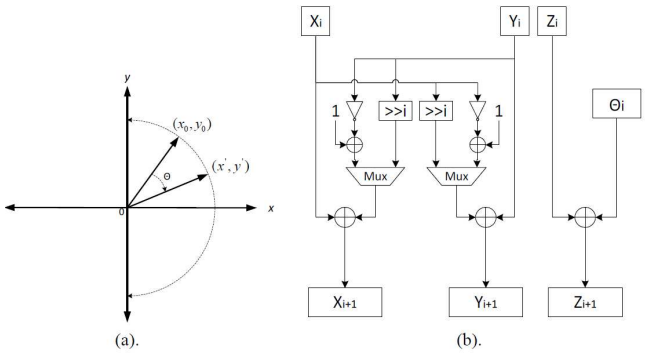

Fig. 7 Illustration of the Circular CORDIC.

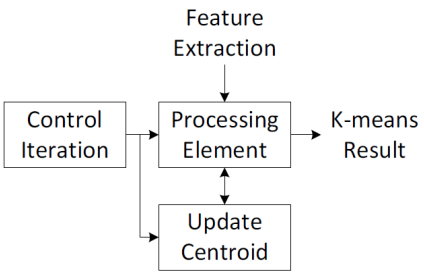

Fig. 8 Block diagram of the K-means.

Now we set two source signals, i.e., X_1 and X_2 and these two source signals are complex numbers generated by FFT module. The equations are formula(10), (11), (12) and (13) where X_1 and X_2 are two input signals. The proposed hardware architecture for feature extraction is showed in Fig. 6. ROM can output the frequency normalization which involves dividing the phase difference by 1 i.e. $1/\alpha$.

The input X_1 and X_2 are both complex numbers, we compute the sita, remainder, square and divider function of input signals with the CORDIC (COordinate Rotation DIgital Computer) architecture [26]. Here sin, cos functions use architecture of fourth-order Maclaurin series. In 'sita, remainder, square and divider function' module uses seven segment linear approximation to produce the value by CORDIC. CORDIC is the acronym of a trigonometric algorithm for Coordinate Rotation Digital Computer. It was first introduced by Jack Volder [27] and later extended by Walther [28]. It only consists of shifts and add. However, it can generate solutions for trigonometric and some transcendental functions. This algorithm is derived from the general provided rotation transform:

$$\begin{cases} X_{i+1} = X_i - \mu d_i Y_i 2^{-i} \\ Y_{i+1} = Y_i - d_i X_i 2^{-i} \\ Z_{i+1} = Z_i - d_i \theta_i \end{cases} \quad (24)$$

CORDIC method can be generalized by introducing a parameter μ and redefining the dedicated angle serials. Through these modifications, algorithm is extend to perform more functions. The graphical representation and architectural mapping for a single step of CORDIC rotation are shown in Fig. 7. Here the circular CORDIC architecture computes trigonometric function and magnitude of a vector whereas the linear mode of CORDIC architecture computes linear functions such as multiplication and division.

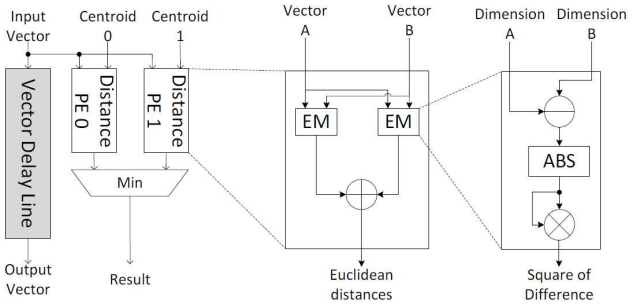

Fig. 9 Architecture of the Processing Element.

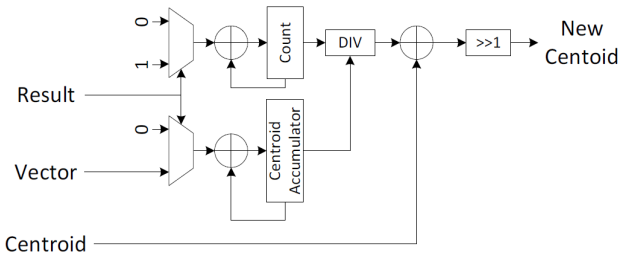

Fig. 10 Architecture diagram of Update Centroid.

4.3 Architecture of Clustering

The proposed k-means hardware architecture is for two dimensional input and two clusters in this work. Fig. 8 shows the block diagram of the proposed k-means module. The data vectors are serial input to the system. In the beginning, we select the first two input vectors as the initial centroids of the two clusters. The input vectors are sent as input to the distance (processing element) PE to compute the distance between input vectors and every centroid. The information of distance in distance PE is compared with the results, which describes the cluster to which input vector belongs. The compared results are added to the summation of the relevant cluster accordingly and fed into update centroid. The update centroid computes the new centroid by dividing the sum values with data count value, and then new centroid is sent back to the PE.

Now, the circuit architecture and computation of the modules in k-means architecture [29] will be introduced, which include PE and update centroid architectures. The design is based on scalability. The scalable design can be easily extended and used for more cluster numbers without designing whole design again.

The architecture for distance computing is important, where we call it as processing element. In PE, and we use Euclidean distances as the distance which calculates the absolute value of the subtraction of two input values to represent the distance between two inputs. The proposed PE have two inputs, where one is the input vector, while the other is the clustering centroid. Fig. 9 shows the architecture of the PE, which includes two distance PEs and the Vector Delay Line. Two distance PEs are used to calculate the distances between input vectors and each of the two centroids to find the nearest centroid of an input vector in one cycle. The "EM" modules in Fig. 9 is capable of computing the Euclidean distance. Two dimensions of input vectors and centroids can be processed simultaneously by this module.

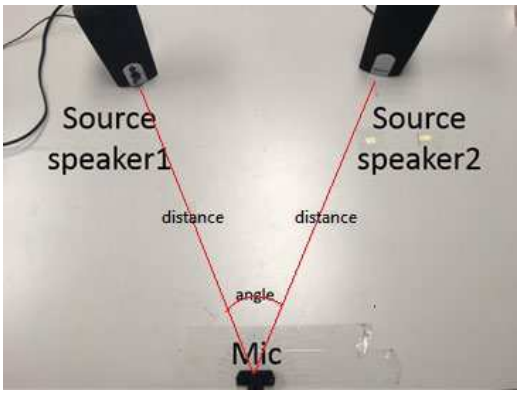

Fig. 11 The whole system design in real environment.

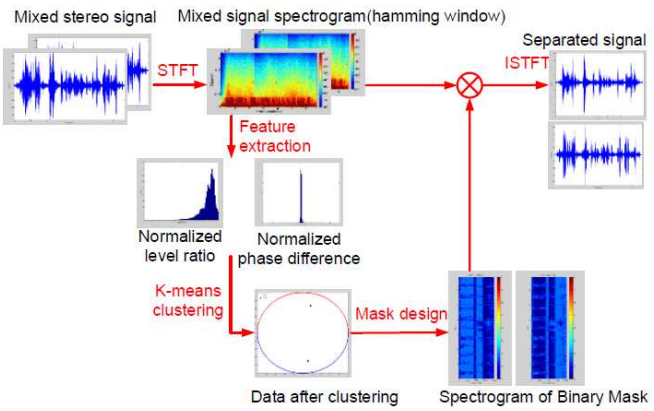

Fig. 12 The procedure of binary mask approach.

Next, the hardware architecture for the Update Centroid is showed in Fig. 10. The module is designed to sum up the vectors in the same cluster and sum up the new centroid. First, Update Centroid add up the value of the vectors in every dimension separately. It can add up values of same dimension with multiplexers. As all of the input vectors in the cluster sum are classified, Update Centroid divides the sum values by the number of the vectors in the cluster. After completion of processing, the module outputs the updated new clustering centroids.

5. Results and Discussion

5.1 Software Experimental Results

The real world experiment environment is shown in Fig. 11. We utilize two omnidirectional microphones with 4.5 cm spacing (Audio-Technica AT9900). The distance between the microphones and sources is 100 cm. The angle between two speakers and microphones is 100°.

The diagram of every process for BSS based on binary mask is shown in Fig. 12. At the beginning, we input a stereo mixture recorded by Audio-Technica AT9900. The sampling rate was 8 kHz and the STFT frame size was 512 with 256 overlapping points. Than we transform stereo mixture signal into time- frequency domain by STFT. After normalize features, we apply *k*-means to clustering. Next, the binary mask is generated by the centroids of the clustered results. Finally, we use binary mask on time- frequency spectrum to separate the mixed signal.

Take the female-male mixture from TIMIT database as an

Table 2. Comparison of original and proposed method at DOA difference 100° for speeches.

	Two speakers	Average iterations for K-means	Average SIR(dB)	Average SDR(dB)
Original Clustering	Female-Female	21.7	14.8	2.8
	Female-Male	26.6	15.5	3.3
	Male-Male	21.2	12.1	1.1
	Average	23.2	14.1	2.4
Proposed Method	Female-Female	15.5	14.3	2.9
	Female-Male	19.4	15.1	3.2
	Male-Male	16.4	12.2	0.9
	Aervage	17.1	13.9	2.3

Table 3. Computational time analysis for binary mask approach

Execution Time (second)	STFT	Feature extraction	k- means	Binary mask	Total
Original	1.87 [20]	0.28 [20]	3.31	0.10	5.56
Proposed	1.83	0.23	2.52	0.10	4.68

Table 4. Comparison of original and proposed method under various positions.

	Two speakers	Average iterations for K-means	Average SIR(dB)	Average SDR(dB)
20°	Original Clustering	33.8	4.9	-2.1
	Proposed Method	21.8	4.5	-2.3
40°	Original Clustering	30.4	12.1	0.7
	Proposed Method	26.6	13.7	0.6
60°	Original Clustering	22.2	12.1	2.9
	Proposed Method	19.6	12.1	2.9
80°	Original Clustering	26.2	12.9	3.4
	Proposed Method	17.8	12.6	3.5
100°	Original Clustering	26.6	15.5	3.3
	Proposed Method	19.4	15.1	3.2
120°	Original Clustering	25	12.6	2.1
	Proposed Method	22	12.6	1.9
140°	Original Clustering	18.2	11.7	2.1
	Proposed Method	22.2	12.4	2.2
160°	Original Clustering	23.8	11.3	0.8
	Proposed Method	18.4	11.7	0.7

example, where we compare the number of iterations for the original *k*- means clustering and the number of iterations of it

In order to compare the result of the original clustering and proposed method for every mixture, we repeat the clustering five times, each with a new set of initial cluster centroid positions. Table 2 shows the typical simulations where the sounds are mixed with several people. These three cases are widely used to demonstrate the feasibility of blind source separation. In Table 2, we can observe that after reducing DOA variance, the iterations are lower.

Referring to the quality issue, there are two criteria to measure the separated result by utilizing BSS_EVAL [30]. One is Signal to Distortion Ratio (SDR) and the other one is Signal to Interference Ratio (SIR), where S_{target} is source signal; e_{interf}, e_{noise}, and e_{artif} are interferences, noise, and artifacts error terms, respectively.

$$\mathrm{SIR} = 10\log_{10}\frac{\left\|s_{target}\right\|^2}{\left\|e_{target}\right\|^2} \qquad (17)$$

$$\mathrm{SDR} = 10\log_{10}\frac{\left\|s_{target}\right\|^2}{\left\|e_{int\,erf} + e_{noice} + e_{artif}\right\|^2} \qquad (18)$$

The SIR determines the ratio of energies of the desired signal and the interference in the separated signal. SIR is highly influenced by filtration of the measured signal, which might be misleading, especially in audio separation. The SDR provides a supplementary criterion of SIR that reflects the difference between the desired and the estimated signal in the mean-square sense. SDR is highly sensitive to the filtering, which may give a rigid evaluation for methods which apply a long separation filter.

The experimental results for SIR and SDR are shown in Table 2. It shows that the proposed method provides almost the same quality when compared with the original BSS.

We further perform some evaluation at aspect of whole system. Table 3 illustrates the computational time analysis for the binary mask approach for STFT, feature extraction, *k*-means, binary mask and ISTFT. For Table 3, 30 seconds source signal is used, and it is averaged over three testing mixtures in terms of different genders. In order to evaluate the performance of different positions between sources and microphones, we built this experiment for the female-male case. In Table 4 and in Fig. 13, different angle of source signal are present, most of the average number of iterations are smaller.

5.2 Hardware Experimental Results

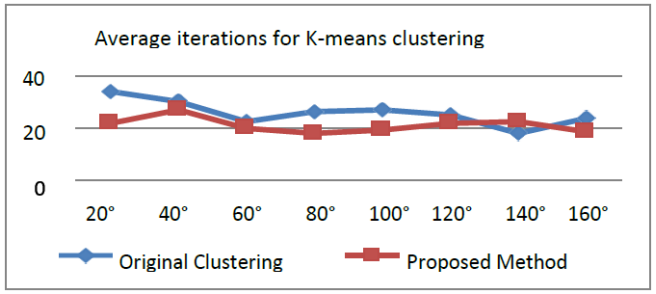

Fig. 13 Comparison of average iterations in K-means clustering for original and proposed method.

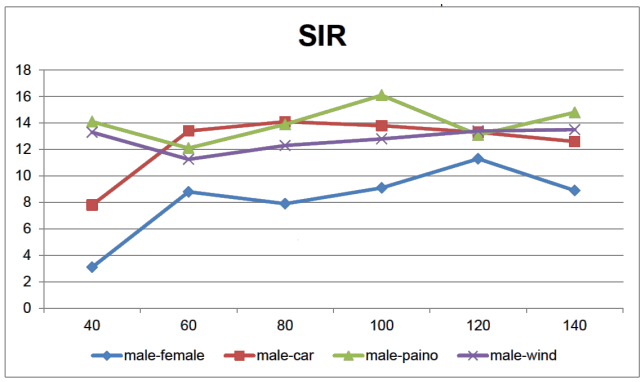

Fig. 14: SIR result for hardware blind source separation

In Fig. 14, the result of the hardware blind source separation for every mixture is shown with different DOA and different noise sources. Totally five different sources are used as noise. The first one, male-female speaking, is a common simulation in BSS. In our noise reduction application, we include more realistic source as noise and simulate them. These sources include car driving, the piano and wind sound. The sampling rate of the source signal is 7350 Hz and the window size is 512 with 256 overlapping points.

This BSS chip is implemented using TSMC 90nm technology and the cell-based design flow. Fig. 15 shows the layout view of the chip. ASIC chip contains 84 I/O pads. We use single-port memory which have 69 Kbits. Total gate count is 119.71k without memory. The total size of core and die is 1.095 x 1.095m^2 and 1.655 x 1.655 m^2 respectively. With power supply of 1.0V, the design achieves 10 MHz; in addition, the power dissipation is 2.92 mW at 10 MHz. Table 5 summarizes the chip specifications.

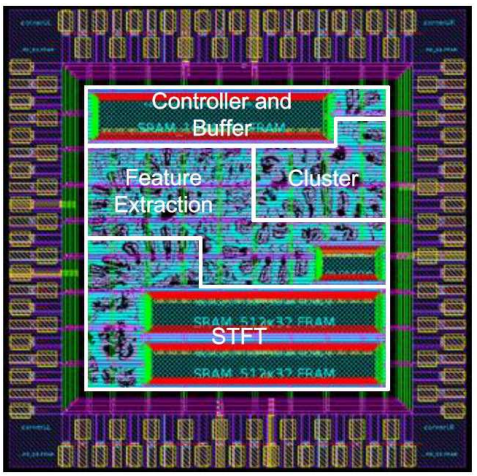

Fig. 15 Chip layout of BSS.

Table 5. Chip Specification of BSS.

Item	Specification
Technology	TSMC 90nm
Voltage	1.0V / 3.3V (Core / IO)
Operation frequency	10MHz
Chip area	1.095 x 1.095m^2
Core area	1.655 x 1.655 m^2
Gate count	119.71K
Memory requirement	69Kbits
Power consumption	2.92 mW
Total pins	84 pins

Table 6 lists several implementation results of BSS systems. To eliminate the process factor, the power consumption of each design has been normalized to 90 *nm* technology by the following equation [31]:

$$Power_{normalized} = Power \times \left(\frac{1.0}{Voltage}\right)^2 \times \left(\frac{90}{Process}\right)$$

$$\text{Throughput} = \frac{sample-per-channel \times second \times 16bit \times 2 \times 2 \times Chip-frequency}{execution-cycle}$$

In the circuit simulation, "second" is the length of the sound signal (second), each of our sample is 16 bits, two "2" represent the overlap 256 and two-channel. Our design achieves frequency of 10MHz. Input of 5 second sound signal needs 4685568 cycles

Table 6. Comparisons of Chip Specification.

	J.C. Wang [32]	L.-D. Van [33]	K.-K. Shyu [34]	C.-M. Kim [35]	This work
Application	Speech	ECG	EEG	Speech	Speech
Algorithm	FastICA	FastICA	ICA	ICA	DOA
Technology	TSMC 90	UMC 90	FPGA	Hynix 0.35um	TSMC 90
Channels	4	8	4	-	2
Samples per channel	-	256	-	-	512
Speed(MHz)	100	100	68	-	10
Power(mW)	54.86	16.35	-	14.5	2.92
Gate count(K)	199	272	315	-	119.71
Memory(bit)	-	68K	24K	42K	69K
Core size	0.54x0.54	1.22x1.22	-	-	1.09x1.09
Power normalized	54.86 mW	16.35 mW	-	0.34 mW	2.92 mW

to process data and execution time is 0.5 seconds. Throughput of the design is 5.0196Mbps.

In Table 6, we compare four chips with our work. In [32], they use FastICA for speech separation and the technology is TSMC90. FastICA is also used by [33]. And in [34], ICA is used for EEG signal separation on FPGA. In the other hand, [35] implemented ICA method on chip with Hynix 0.35um technique.

6. Conclusions

In this paper, a fast binary mask approach for blind source separation is proposed. First, we input the time domain signal and transform it to the time-frequency domain with STFT. Second, we utilize feature extraction to obtain geometric information and the formula. With the aid of the reduction of DOA variance, we decrease the variance of features in order to obtain lower iterations of *k*-means clustering. Finally, according to the clustered features, a time frequency mask is generated. Our method does not require any prior information or parameter estimation. Experimental results with various mixtures are simulated in real environments to verify the effectiveness of the proposed method. We further perform the experiment on real time application which is based on source separation situation and acting as a noise reduction system. It is worth mentioning that the proposed method has low average number of iterations and low total sum of distance. Furthermore, the performance is slightly improved in terms of average SIR and SDR. Finally, hardware simulation of the proposed chip is performed by Verilog language.

Abbreviations

BSS: Blind source separation

ICA: Independent Component Analysis

DOA: Direction of Arrival

MAP: Maximum a Posterior

FFT: Fast Fourier transform

CORDIC: COordinate Rotation DIgital Computer

SDR: Signal to Distortion Ratio

SIR: Signal to Interference Ratio

Declarations

CONSENT FOR PUBLICATION
Not applicable.

AVAILABILITY OF DATA AND MATERIAL
Please contact author for data requests.

COMPETING INTERESTS
The authors declare that they have no competing interests.

FUNDING
Not applicable.

Authors' Contributions
Pei-Yun Liu conceived and designed the study. Yu-He Chiou performed the experiments. Tsung-Han Tsai reviewed and edited the manuscript. All authors read and approved.

Acknowledgements
Not applicable.

References

[1] Z. Koldovský ˇ; P. Tichavský, "Time-Domain Blind Separation of Audio Sources on the Basis of a Complete ICA Decomposition of an Observation Space," Audio, Speech, and Language Processing, IEEE Transactions on, vol. 19, pp. 406-416, Feb. 2011.
[2] H. Saruwatari; K. Sawai; T. Nishikawa; A. Lee; K. Shikano; A. Kaminuma; M. Sakata; D. Saitoh,"Speech Enhancement Based on Blind Source Separation in Car Environments," Data Engineering Workshops 21st International Conference on, Apr. 2005.
[3] Araki, S., Sawada, H., Mukai, R. and Makino, S., "Normalized observation vector clustering approach for sparse source separation," In Proceedings of the EUSIPCO, 2006.
[4] WANG Miao; CAI Xiao-xia; ZHU Ke-fan, "A Blind Separation of Variable Speed Frequency Hopping Signals based on Independent Component Analysis," ITNEC, 2019.
[5] S. Faiz Minhas; P. Gaydecki, "A hybrid algorithm for blind source separation of a convolutive mixture of three speech sources", EURASIP Journal on Advances in Signal Processing, Jun. 2014.
[6] Nobutaka Ito; Shako Araki; Takuya Yoshioka; Tomohiro Nakatani , "Relaxed disjointness based clustering for joint blind source separation and dereverberation," IWAENC, 2014
[7] Q. PanEmail; T. Aboulnasr, "Time-Domain Convolutive Blind Source Separation Employing Selective-Tap Adaptive Algorithms", EURASIP Journal on Audio, Speech, and Music Processing, Apr., 2007.
[8] M. Gholamrezaii, M. R. Aghabozorgi and H. R. Abutalebi, "Blind separation of speech target sources using ICA in the frequency domain," 2010 5th International Symposium on Telecommunications, Tehran, pp. 765-768, 2010.
[9] Robledo-Arnuncio; E.; Biing-Hwang Juang, "Issues in frequency domain blind source separation - a critical revisit", Acoustics, Speech, and Signal Processing, IEEE International Conference on (ICASSP), vol.5, Mar. 2005.
[10] S. Winter; W. Kellermann; H. Sawada; S. Makino1,"MAP based underdetermined blind source separation of convolutive mixtures by hierarchical clustering and L1-norm minimization", EURASIP Journal on Advances in Signal Processing, 2007.
[11] S. Winter , H. Sawada , S.Araki , S. Makino,"Overcomplete BSS for convolutive mixtures based on hierarchical clustering", Independent Component Analysis and Blind Signal Separation Lecture Notes in Computer Science, vol. 3195, pp. 652-660, 2004.
[12] M. Aoki; M.Okamoto; S. Aoki; H. Matsui; T. Sakurai; Y. Kaneda, "Sound source segregation based on estimating incident angle of each frequency component of input signals acquired by multiple microphones," Acoustical Science and Technology, pp.149-157, Jan. 2001.
[13] I. JafariEmail; S. Haque; R. Togneri; S. Nordholm,

"Evaluations on underdetermined blind source separation in adverse environments using time-frequency masking," EURASIP Journal on Advances in Signal Processing, Oct., 2013.
[14] C. Kang, W. Fan, X. Zhang and J. Li, "A kind of method for direction of arrival estimation based on blind source separation demixing matrix," 2012 8th International Conference on Natural Computation, Chongqing, pp. 134-137, 2012.
[15] H. Sawada; R. Mukai ; S. Araki ; S. Makino," A robust and precise method for solving the permutation problem of frequency-domain blind source separation", IEEE transactions on speech and audio processing, vol. 12, no. 5, Sep. 2004.
[16] A. R. Katti, A. P. B. K. Shakeeb and P. A. M. Vijay, "Novel VLSI architecture for real-time blind source separation," 3rd International Conference on Advances in Recent Technologies in Communication and Computing (ARTCom 2011), Bangalore, 2011, pp. 209-213.
[17] J. Wang et al., "VLSI Design for Convolutive Blind Source Separation," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 63, no. 2, pp. 196-200, Feb. 2016.
[18] M. Stanaćević, S. Li and G. Cauwenberghs, "Micropower Mixed-Signal VLSI Independent Component Analysis for Gradient Flow Acoustic Source Separation," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 63, no. 7, pp. 972-981, July 2016.
[19] P. Bofill; M. Zibulevsky, "Blind separation of more sources than mixtures using sparsity of their short-time Fourier transform", in Proceedings of the ICA2000, pp. 87-92, 2000.
[20] Jourjine, A. ; Rickard, Scott ; Yilmaz, O. "Blind separation of disjoint orthogonal signals: demixing N sources from 2 mixtures",Acoustics, IEEE International Conference on Speech, and Signal Processing (ICASSP) , vol. 5, 2000.
[21] Guy J. Brown, D. Wang, "Separation of Speech by Computational Auditory Scene Analysis", Speech Enhancement, pp. 371-402, 2005.
[22] S.Arakia; H. Sawadaa; R. Mukaia; S. Makinoa ,"Underdetermined blind sparse source separation for arbitrarily arranged multiple sensors", Signal Processing, pp.1833-1847, Aug. 2007.
[23] B.-C. Lin, Y.-H. Wang, J.-D. Huang, J. Y. Jou, "Expandable MDC-based FFT architecture and its generator for high-performance applications," IEEE International SOC Conference, pp. 188-192, Sept. 2010.
[24] J.-C. Kuo, C.-H. Wen, C.-H. Lin, A.-Y. Wu, "VLSI design of a variable-length FFT/IFFT processor for OFDM-based communication systems," EURASIP Journal on Applied Signal Processing 2003, pp. 1306–1316, Dec. 2003.
[25] S.-F. Lei, S.-N. Yao, "A memory-free modified discrete cosine transform architecture for MPEG-2/4 AAC," IET Circuits, Devices and Systems, vol. 4, pp. 14-23, Jan. 2010.
[26] B. Yang, D. Wang, L. Liu, "Complex division and square-root using CORDIC," 2012 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet), pp. 2464-2468, Apr. 2012
[27] J. VOLDER, "Binary computation algorithms for coordinate rotation and function generation," Convair Report IAR-1 148 Aeroelectronics Group, 1956.
[28] J. S. Walther,"A unified algorithm for elementary functions," in Spring Joint Comp, ser. Conference 1971, pp. 379–385. 1971
[29] T.-W. Chen, S.-Y. Chien, "Flexible hardware architecture of hierarchical k-means clustering for large cluster number," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 19, no. 8, pp.1336-1345, Aug. 2011.
[30] Vincent, E. ; Gribonval, R. ; Fevotte, C. "performance measurement in blind audio source separation", Audio, Speech, and Language Processing, IEEE Transactions on, vol.14 ,pp.1462-1469, Jul. 2006.
[31] J.-M. Lin, H.-Y. Yu, Y.-J. Wu, H.-P. Ma, "A Power Efficient Baseband Engine for Multiuser Mobile MIMO-OFDMA Communications," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 57, no. 7, pp. 1779-1792, Jul. 2010.
[32] J.-C. Wang, C.-Y. Wang, T.-C. Tai, M. Shih, S.-C. Huang, Y.-C. Chen, Y.-Y. Lin, L.-X. Lian, "VLSI Design for Convolutive Blind Source Separation," IEEE Transactions on Circuits and Systems II: Express Briefs, vol.63, no. 2, pp. 196-200, Feb. 2015.
[33]L.-D. Van, D.-Y. Wu, C.-S. Chen, "Energy-Efficient FastICA Implementation for Biomedical Signal Separation," IEEE Transactions on Neural Networks, vol. 22, no. 11, pp. 1809-1822, Nov. 2011.
[34] Wei-Chung Huang, Shao-Hang Hung, Jen-Feng Chung, Meng-Hsiu Chang, Lan-Da Van and Chin-Teng Lin, "FPGA implementation of 4-channel ICA for on-line EEG signal separation," 2008 IEEE Biomedical Circuits and Systems Conference, Baltimore, MD, 2008, pp. 65-68.
[35]C.-M. Kim, H.-M. Park, T. Kim, Y.-K. Choi, S.-Y. Lee, "FPGA implementation of ICA algorithm for blind signal separation and adaptive noise canceling," IEEE Transactions on Neural Networks, vol. 14, no. 5, pp.1038-1046, Sep. 2003.

Figures

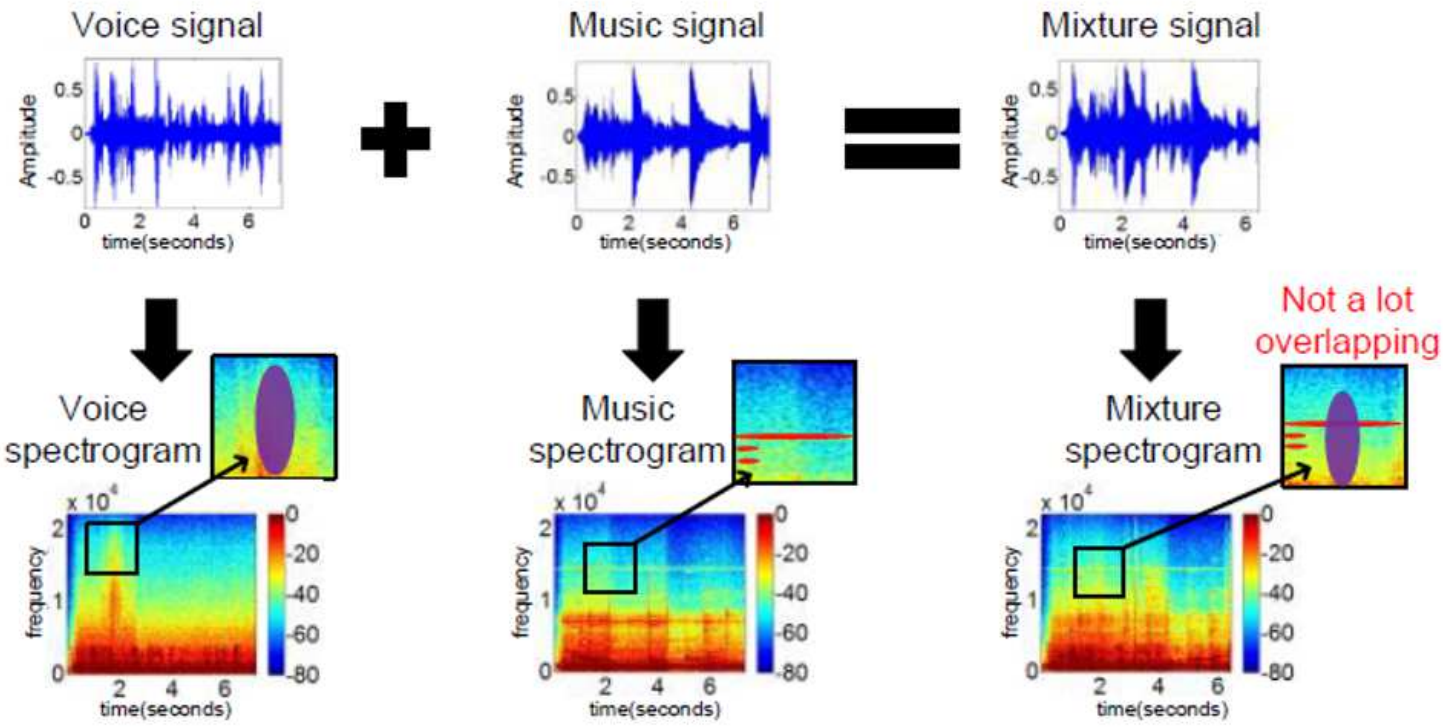

Figure 1

Time frequency spectrum of mixed input signal.

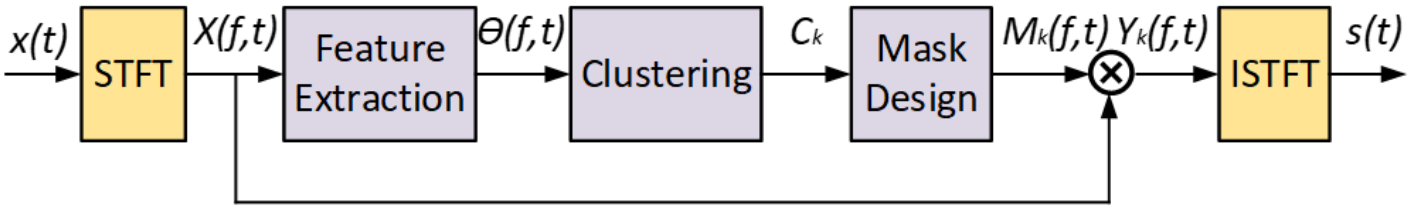

Figure 2

Functional block of binary mask approach.

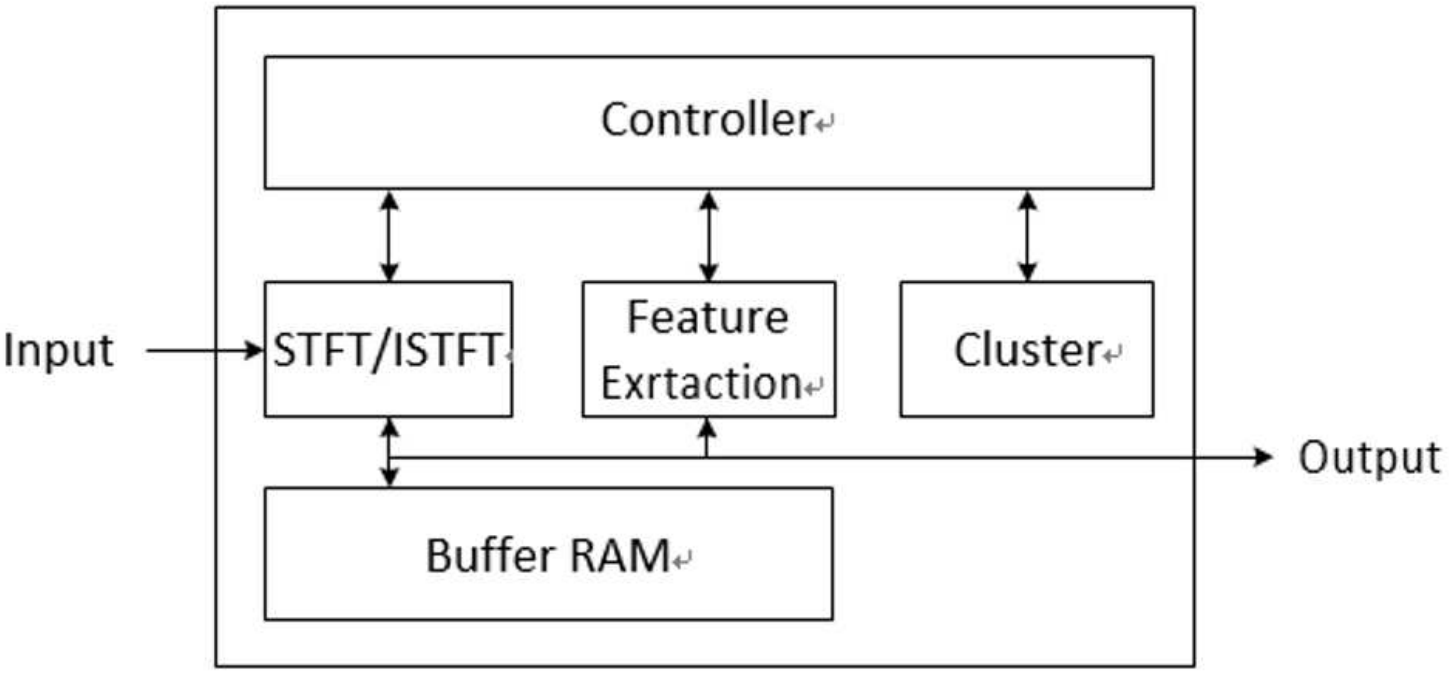

Figure 3

Architecture diagram of BSS.

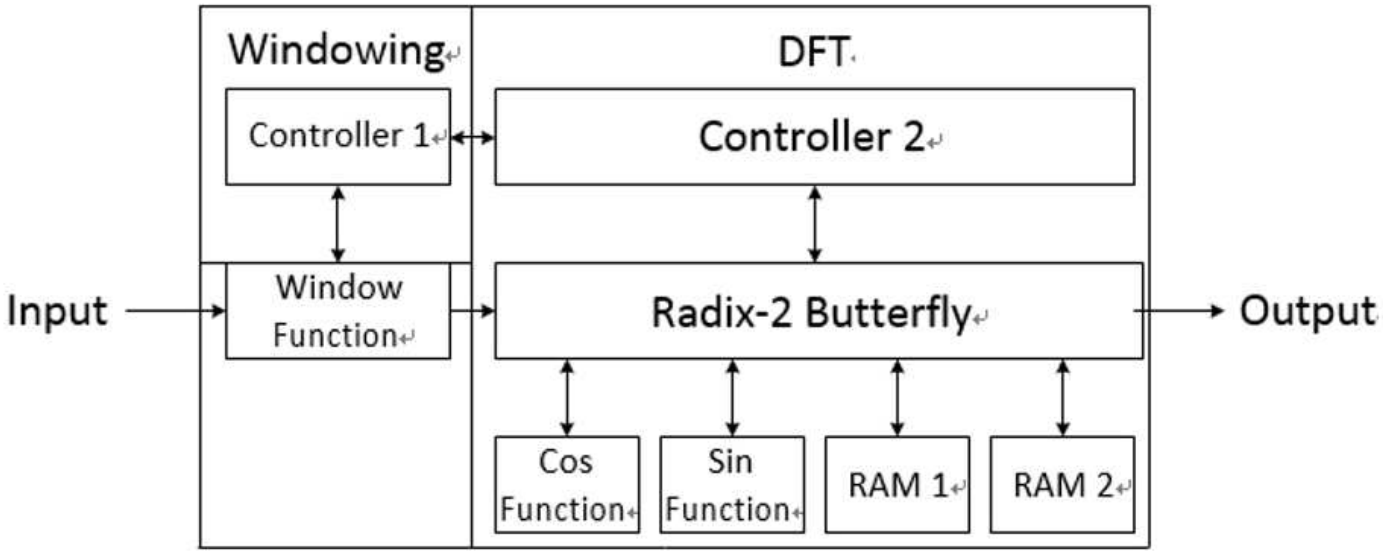

Figure 4

Architecture diagram of STFT.

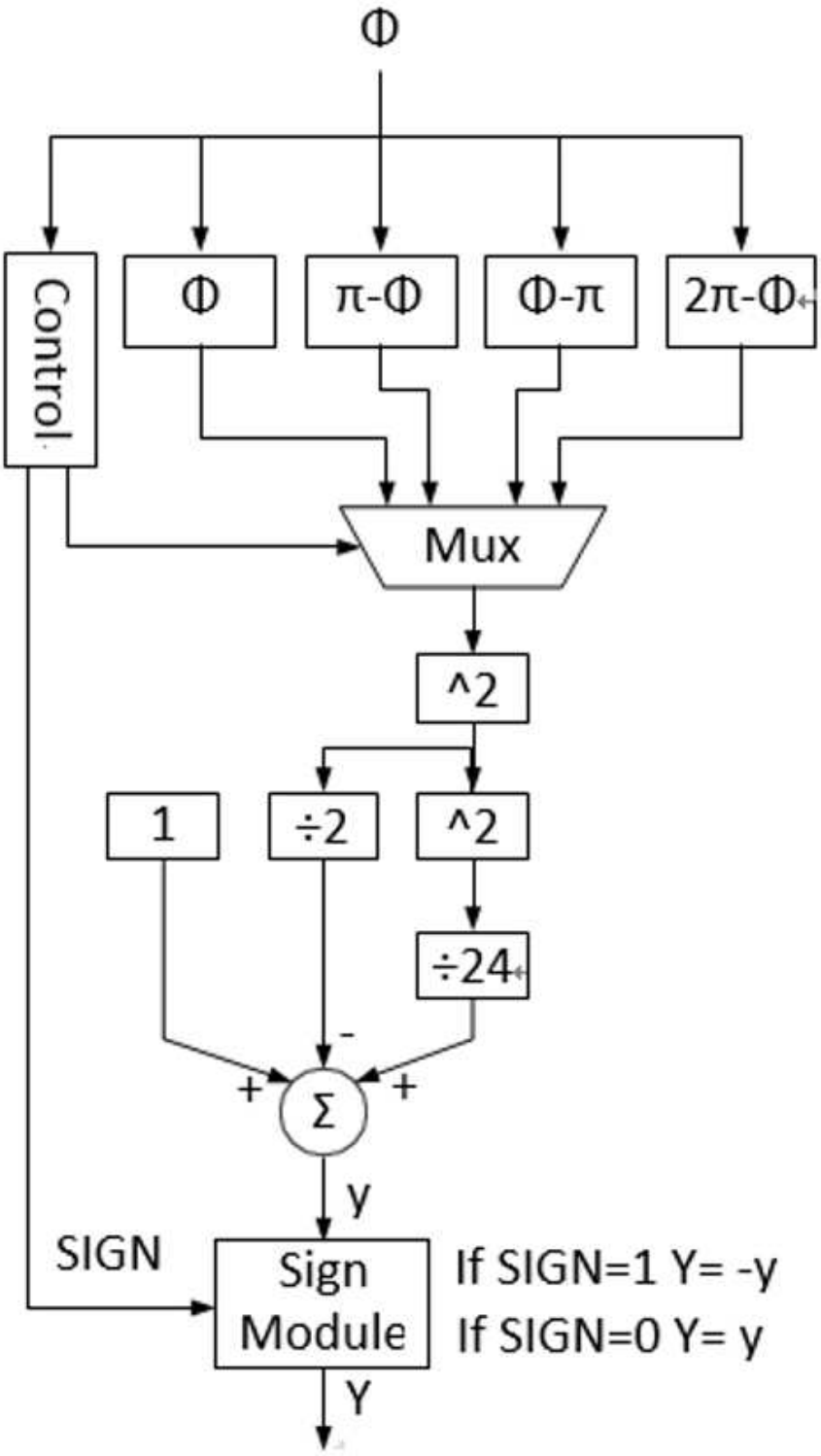

Figure 5

Architecture diagram of fourth-order Maclaurin series.

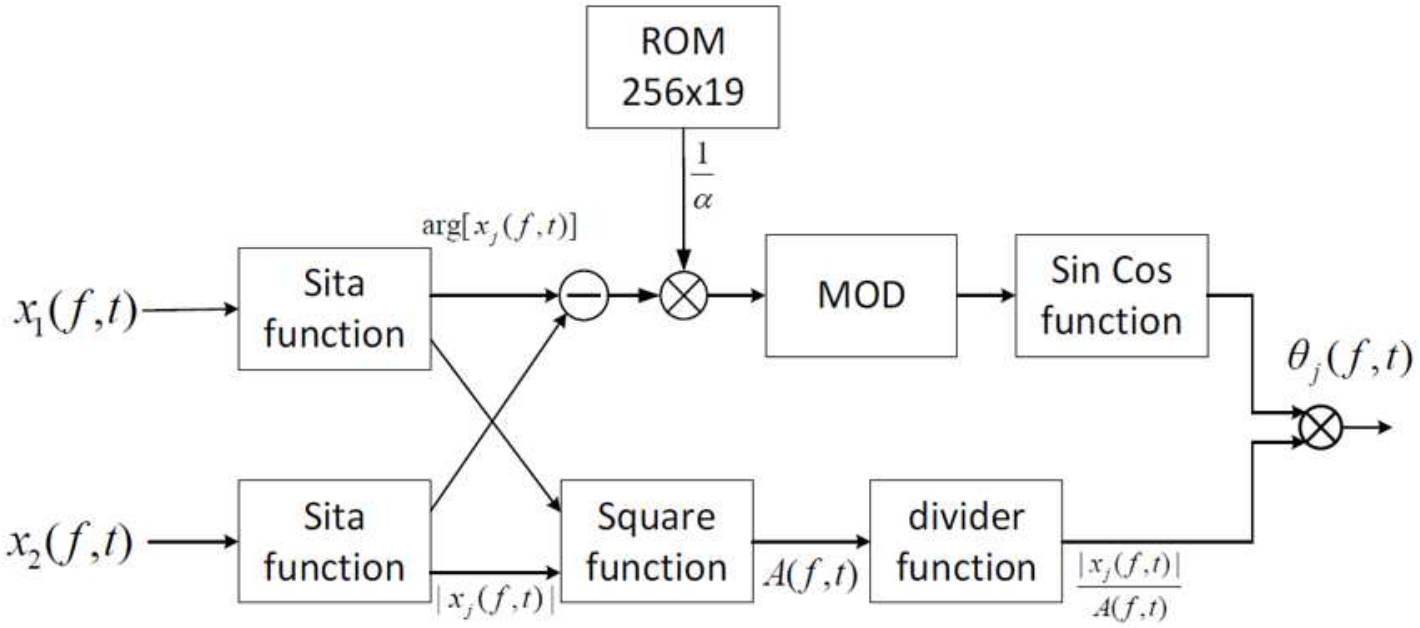

Figure 6

Architecture diagram of feature extraction.

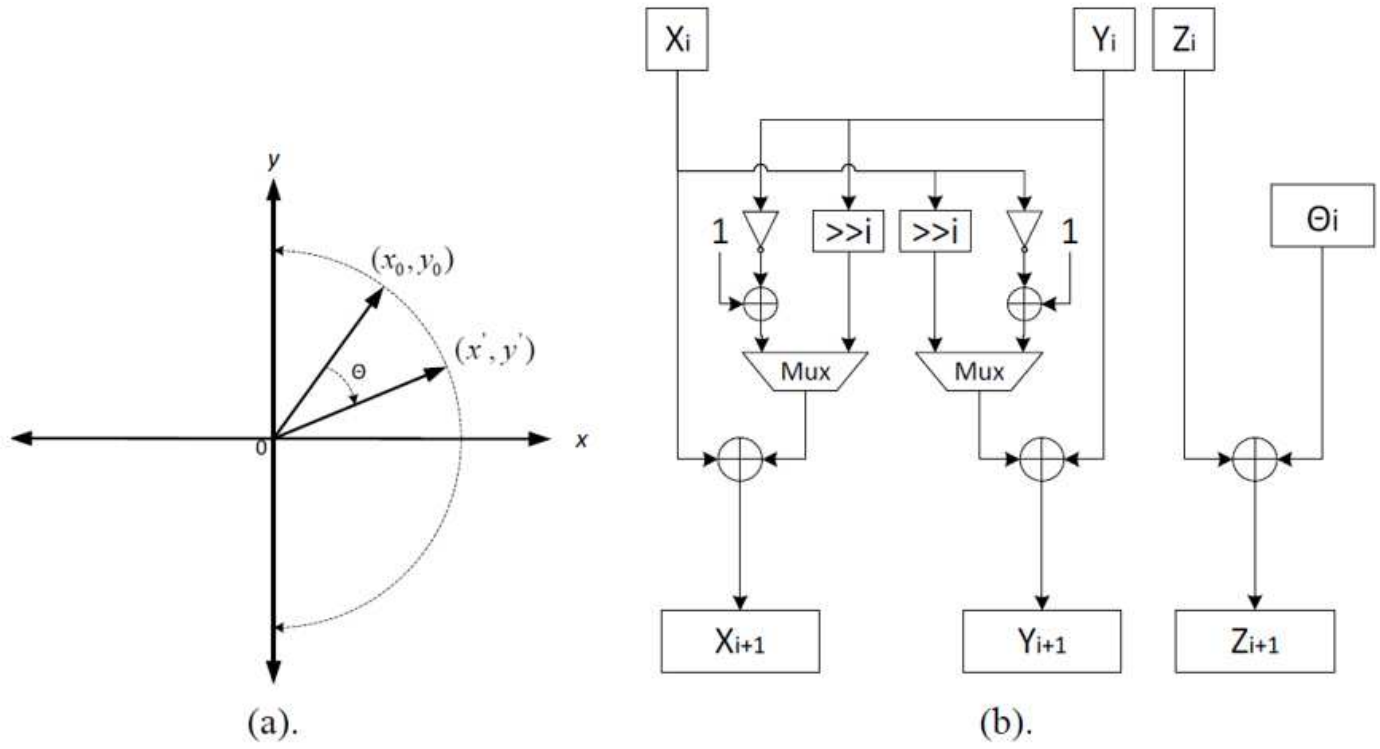

Figure 7

Illustration of the Circular CORDIC.

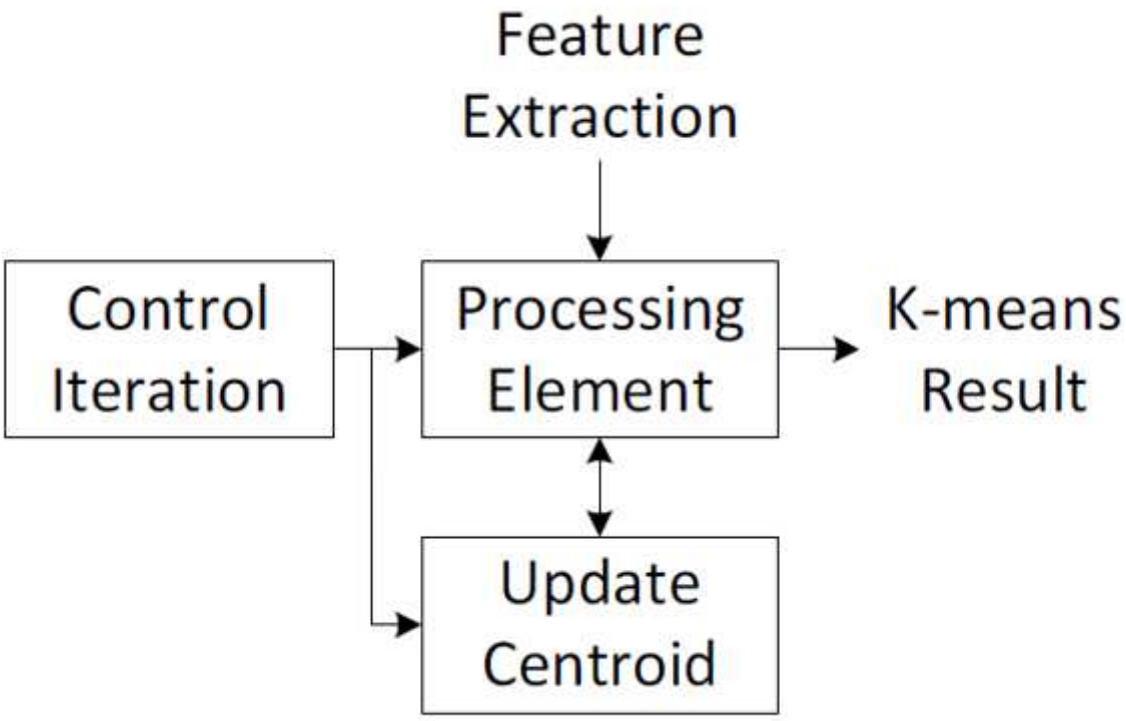

Figure 8

Block diagram of the K-means.

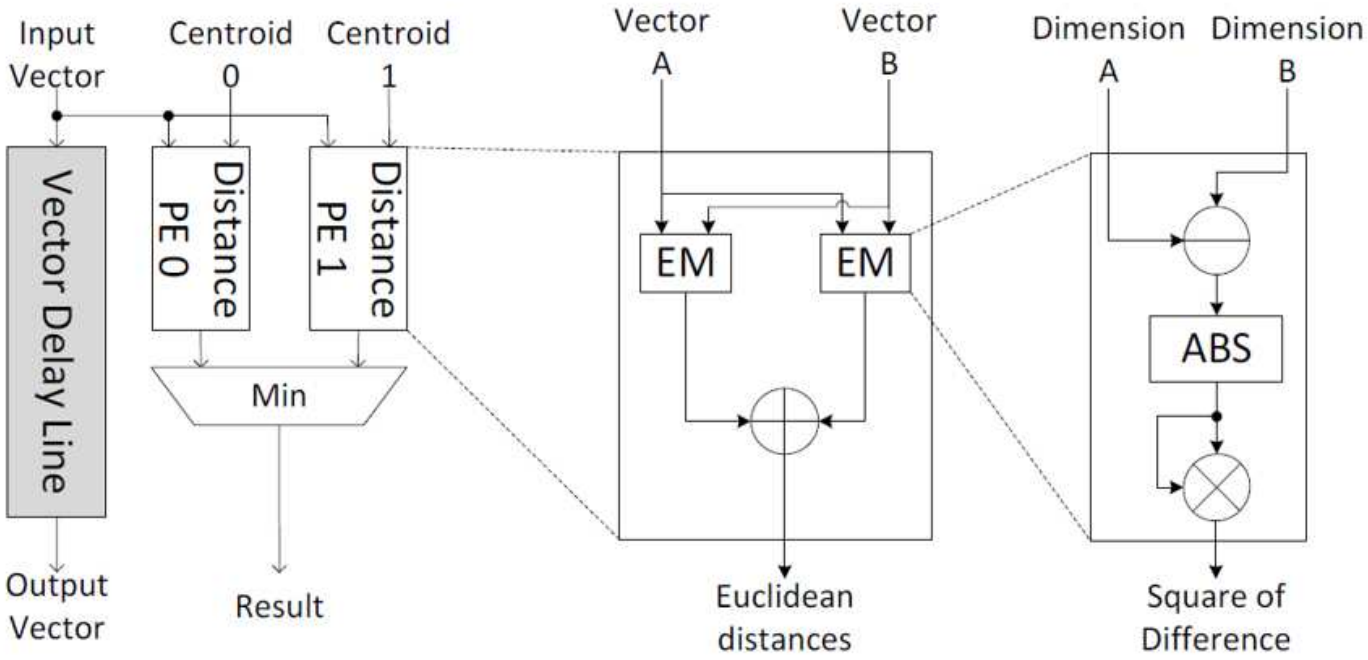

Figure 9

Architecture of the Processing Element.

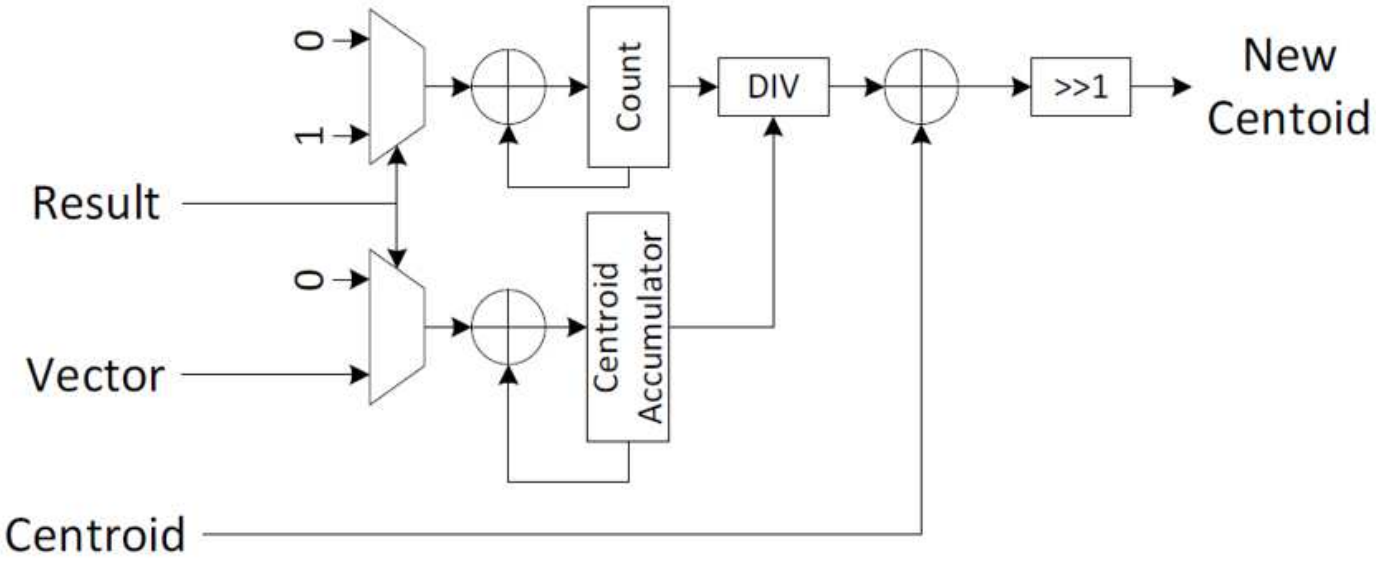

Figure 10

Architecture diagram of Update Centroid.

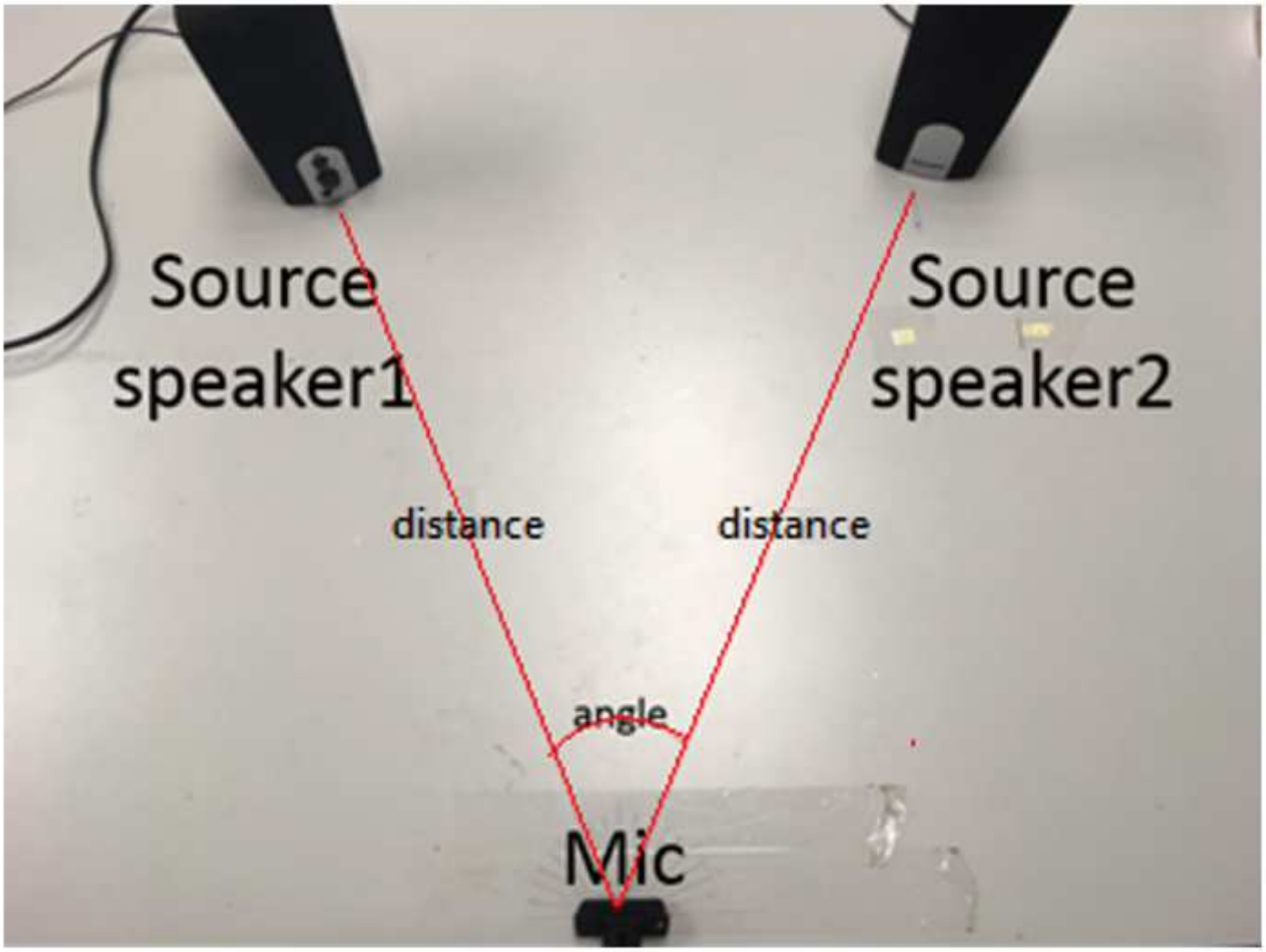

Figure 11

The whole system design in real environment.

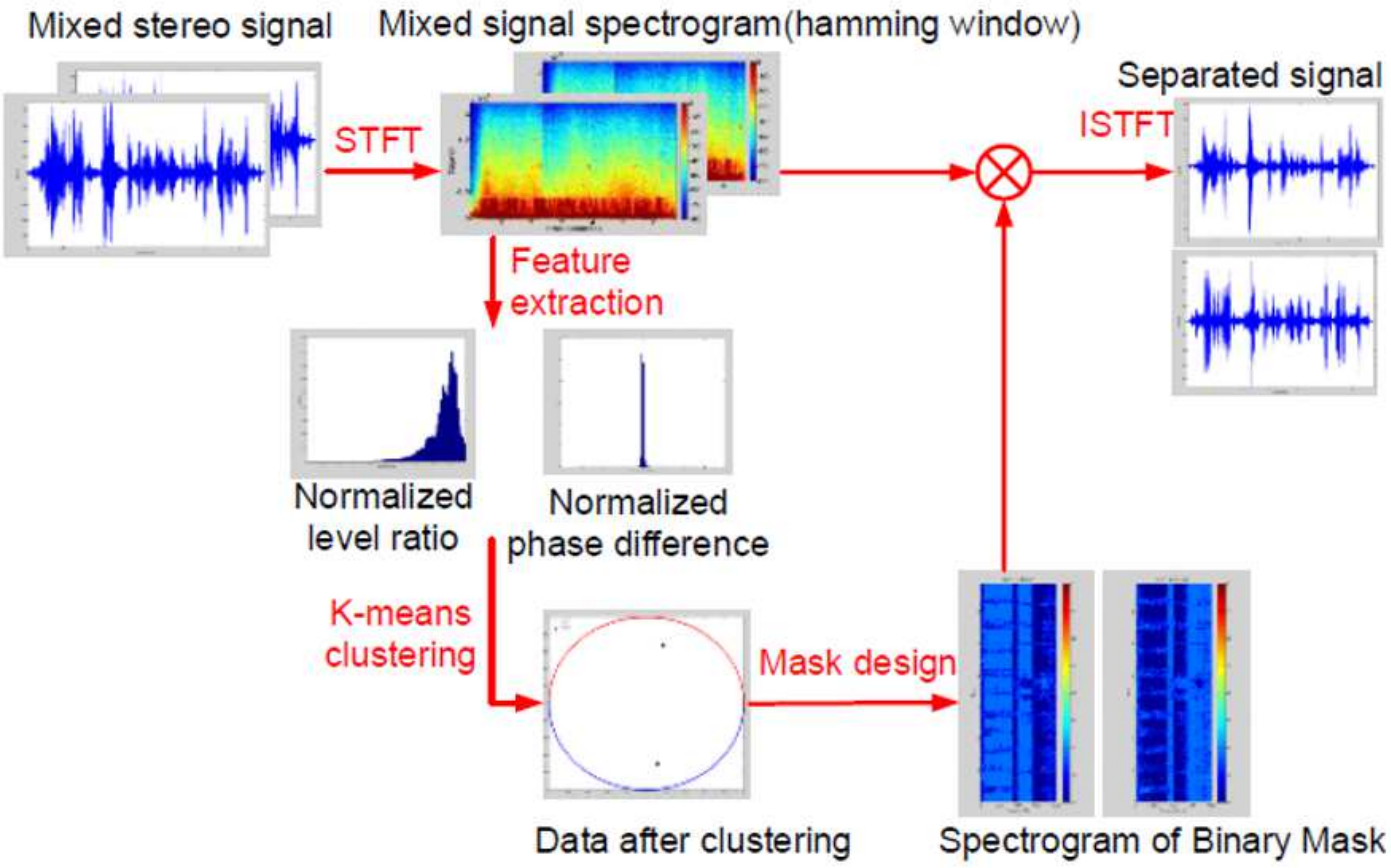

Figure 12

The procedure of binary mask approach.

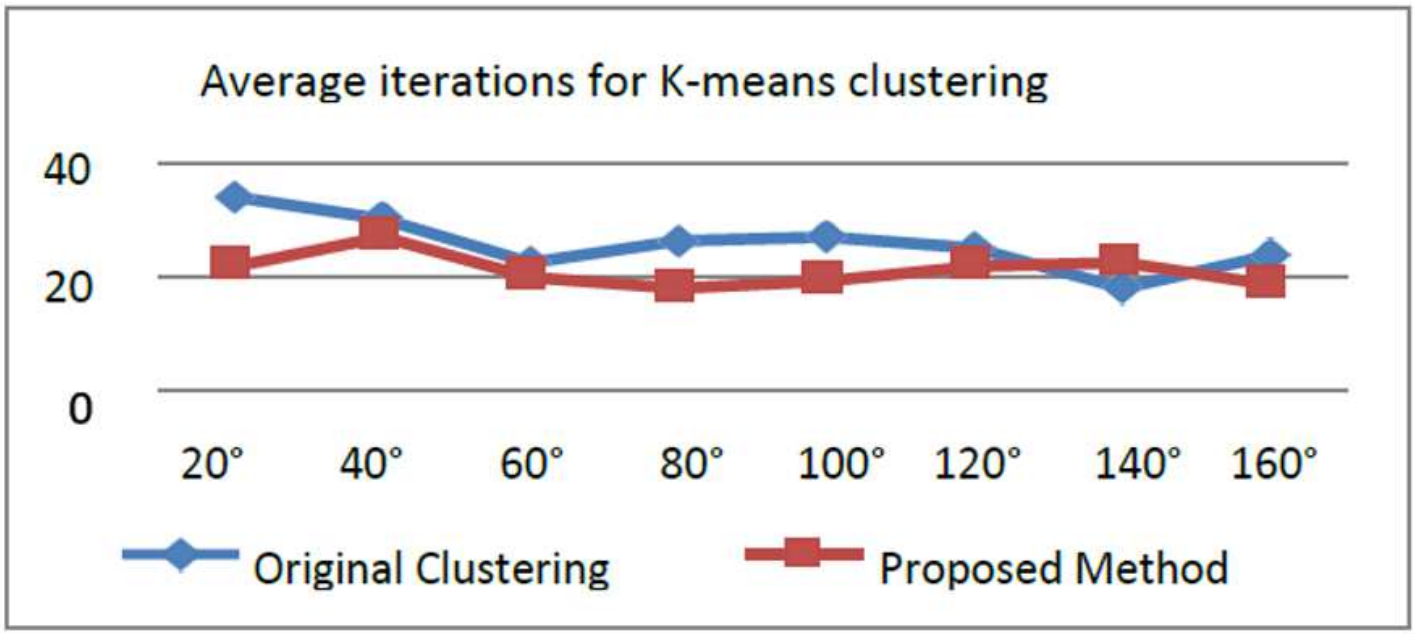

Figure 13

Comparison of average iterations in K-means clustering for original and proposed method.

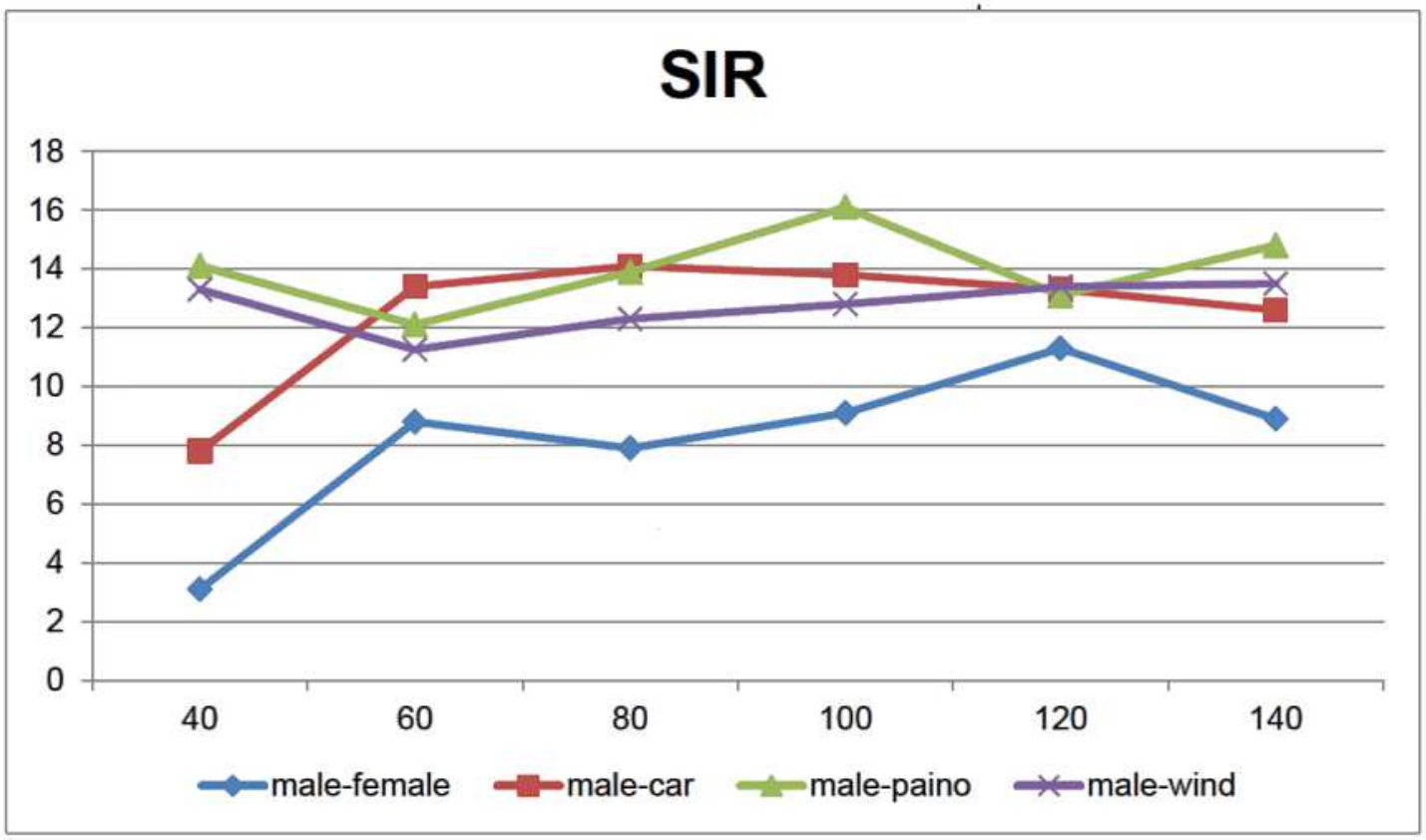

Figure 14

SIR result for hardware blind source separation

Figure 15

Chip layout of BSS.