IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 28, 2019, accepted June 18, 2019, date of publication June 21, 2019, date of current version July 9, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2924340

EERA-KWS: A 163 TOPS/W Always-on Keyword Spotting Accelerator in 28nm CMOS Using Binary Weight Network and Precision Self-Adaptive Approximate Computing

BO LIU[1], (Member, IEEE), ZHEN WANG[2], HU FAN[1], JING YANG[1], BO LIU[1], WENTAO ZHU[1], LEPENG HUANG[1], YU GONG[1], WEI GE[1], AND LONGXING SHI[1], (Senior Member, IEEE)

[1]National ASIC System Engineering Technology Research Center, Southeast University, Nanjing 210096, China
[2]Nanjing Prochip Electronic Technology Company Ltd., Nanjing 210001, China

Corresponding author: Bo Liu (liubo_cnasic@seu.edu.cn)

This work was supported in part by the National Science and Technology Major Project under Grant 2018ZX01031101-005, and in part by the National Natural Science Foundation of China under Grant 61404028 and Grant 61574033.

ABSTRACT This paper proposed an energy-efficient reconfigurable accelerator for keyword spotting (EERA-KWS) based on binary weight network (BWN) and fabricated in 28-nm CMOS technology. This keyword spotting system consists of two parts: the feature extraction based on melscale frequency cepstral coefficients (MFCC) and the keywords classification based on a BWN model, which is trained through the Google's Speech Commands database and deployed on our custom. To reduce the power consumption while maintaining the system recognition accuracy, we first optimize the MFCC implementation with approximate computing techniques, including Pre-emphasis coefficient transformation, rectangular Mel filtering, Framing and FFT optimization. Then, we propose a precision self-adaptive reconfigurable accelerator with digital-analog mixed approximate computing units to process the BWN efficiently. Based on the SNR prediction of background noise and post-detection of network output confidence, the BWN accelerator data path can be dynamically and adaptively reconfigured as 4, 8, or 16 bits. For the BWN accelerator, we proposed a time-delay based addition unit to process bit-wise approximate computing for the convolution layers and fully connected layers, and a LUT based unit for the activation layers. Implemented under TSMC 28 nm HPC+ process technology, the estimated power is 77.8 μW $\sim$ 115.9μW, the energy efficiency can achieve 163 TOPS/W, which is over 1.8× better than the state-of-the-art architecture.

INDEX TERMS Keyword spotting, binary weight network, approximate computing.

I. INTRODUCTION

The keyword spotting (KWS) system is used to automatically detect several particular keywords from a continuous stream of speech and has been utilized in many human-computer interaction applications, such as wearable devices, the Internet of Things (IoT), and so on. In recent years, deep neural networks (DNNs) have been shown to outperform traditional models (i.e., Hidden Markov models and Gaussian mixture models) on a variety of speech recognition benchmarks by a large margin [1][2]. Low power consumption and extreme energy efficiency are very critical for deployments of DNNs for KWS, because they are typically embedded in consumer devices with limited computational resources and energy consumption, such as smart-phones or IoT sensors. As the large amount of neurons and synapses in DNNs incur intensive computation and power consumption, it is still a big challenge to process and accelerate DNNs with high energy efficiency for KWS under various scenarios with different noise types and SNRs.

To overcome the challenge, many DNN accelerators for speech recognition have been proposed in the past decades. These works can be mainly classified into two types: the digital architectures and the analog architectures. For the digital architectures, FPGAs, customized DSPs, ASICs and coarse-grained reconfigurable architectures (CGRAs) are

The associate editor coordinating the review of this manuscript and approving it for publication was Vincenzo Conti.

frequently utilized to implement DNN accelerators. However, FPGAs and DSPs are not suitable for small, low-power applications such as KWS because of their large footprint and power consumption; ASICs can provide the best performance and energy efficiency for specific applications, but it is not desirable for processing DNNs, because they are implemented with predefined function modules and can not suit new design requirements or changes in DNN structures or layer settings. In the last few years, CGRAs have been introduced as an alternative to conventional designs with high flexibility and energy efficiency while accelerating DNNs. Shah M., et al. proposed a low cost DNN structure with fixed weight bit width for 10 keywords detection [3]. To accelerate the DNN effectively, a CGRA accelerator has been implemented under 40nm TSMC technology to accelerate the DNN, and the power consumption is 11.12 mW for KWS. Price M., et al. presented an ultra-low power speech recognizer for both KWS and complex speech recognition tasks, where the adopted DNN is made up of three fully connected (FC) layers and the bit width of data and weight are both 16 bits. This work can reduce the power to 7.78 mW @40MHz with WER of 8.78% under TSMC 65nm low-power logic process [4]. In work [5], a CGRA based DNN accelerator is proposed for mobile intelligence. This DNN accelerator can support voice wake-up function (one keyword recognition) with power consumption of 321 μW. Yin S., et al. proposed a CGRA named Thinker, which can support variant bit widths computing of DNNs, and achieved 1.27 TOPS/W in energy efficiency [6]. In work [7], the authors first presented a speech recognition system based on a optimized Binary Neural Network (BNN), where the bit width of data and weight are both 1 bit. In the computation of this BNN, 99% of operations are additions, and the multiplication operations are almost eliminated. To further improve the energy efficiency, they proposed an ultra-low power CGRA accelerator with digital approximate addition units to process the calculation of each layer in the BNN. For low background noise (SNR $\geq$ 5dB), this work can support real-time KWS with power consumption of 141 μW and can achieve 90 TOPS/W in energy efficiency. However, limited by the low recognition accuracy of BNN, this work can only support one keyword recognition with low background noise.

Analog architectures have also been commonly used for DNN accelerators to implement the processing components of neurons and synapses because of the natural similarity to biological systems. However, the unreliability in the analog system is still a difficult problem to be solved. Besides, they also lack system flexibility and adaptability. For example, Badami, et al. proposed an analog DNN architecture for speech/non-speech classification in a voice activity detection system [8]. Comparisons show that the use of analog analytics brings increased energy efficiency by 10×. However, due to the limitation of analog circuit characteristics, it can only work well under specific conditions, such as specified SNR/database.

In this paper, we proposed an energy-efficient reconfigurable accelerator for keyword spotting (EERA-KWS) based on binary weight network (BWN) using precision self-adaptive approximate computing. This keyword spotting system consists of two parts: the feature extraction based on Melscale Frequency Cepstral Coefficients (MFCC) and the keywords classification based on a BWN model, which is trained through the Google's Speech Commands database and deployed on our custom. The proposed KWS system can support 10 keywords recognition under different noise types (database: Babble, White, Pink) and SNRs (from −5dB to 10dB). The chosen keywords in this work are Google speech commands "Yes", "No", "Up", "Down", "Left", "Right", "On", "Off", "Stop", "Go", along with "silence" and"unknown". Firstly, we present a feature extraction module based on an optimized MFCC with approximate computing techniques as the following: 1. We remove several modules from the traditional standard MFCC without reducing the recognition accuracy for our KWS system; 2. We use shift and addition operations to replace the multiplication operations in the Pre-emphasis module; 3. We use rectangular bandpass filters to replace the triangular bandpass Mel filters in Mel-filtering module; 4. We use FFT operation with no frame overlapping to reduce the computation. Compared to the standard MFCC which is always used for complex speech recognition, with the optimized MFCC, the required computation can be effectively reduced to less than 40% of the standard MFCC, while the recognition accuracy of the proposed KWS system will not be reduced. Secondly, to accelerate the BWN and make it energy efficient, we proposed a precision self-adaptive reconfigurable DNN accelerator with digital-analog mixed approximate computing units. The DNN accelerator can be reconfigured to process the proposed BWN with various layer sizes and data bit width for different computing accuracy requirements. Based on SNR prediction of background noise and post-detection of network output confidence, the DNN accelerator data path can be dynamically and adaptively reconfigured as 4, 8 or 16 bits. Besides, we proposed a self-adaptive linear piecewise calculation method using LUT units to process the activation layers of BWN with reduced computing complexity and power consumption. To further reduce the power consumption of processing BWN, we proposed a time-delay based addition unit to process bit-wise approximate computing for the convolution layers and fully connected layers of BWN. With the proposed precision self-adaptive computing and digital-analog mixed computing, the computing energy consumption can be significantly reduced, compared to those with fixed data bit width and standard computing units.

This paper makes the following contributions:

1)We present a keyword spotting system using an optimized MFCC for feature extraction and a BWN for keywords classification. The MFCC is optimized with several approximate computing techniques including: removing several redundant modules from the traditional MFCC for the proposed KWS system, using shift and addition operations to replace the multiplication operations in the Pre-emphasis module, using rectangular bandpass filters to replace the triangular bandpass Mel filters in Mel-filtering module, and using FFT operation with no frame overlapping. With the optimized MFCC, the required computation can be effectively reduced to less than 40% of the standard MFCC, while the recognition accuracy of the proposed KWS system can remain unaffected.

2)We propose a precision self-adaptive reconfigurable DNN accelerator for the BWN used for keywords classification. We proposed a precision control method based on SNR prediction of background noise and post-detection of network output confidence. With this precision control method, the data path of the proposed DNN accelerator can be dynamically and adaptively reconfigured as 4, 8 or 16 bits, and the activation layers of BWN can be computed by self-adaptive linear piecewise calculation, for various scenarios with different noise types and SNRs.

3)We propose a time-delay based addition unit architecture and its circuit implementations for DNN approximate computing. The proposed time-delay based addition unit can be dynamically reconfigured to adapt to different data bit width and computing accuracy requirements. This approximate computing unit can contribute a significant decrease in energy consumption compared to that with standard computing units. Implemented under TSMC 28nm technology, our work can achieve 163 TOPS/W in energy efficiency, which is over 1.8× better than the state-of-the-art architecture.

The rest of this paper is organized as follows. Some related preliminary works are briefly discussed in section II. Section III describes the KWS prototype system, the BWN accelerator and the optimized MFCC module. In section IV, we proposed the energy-efficient approximate computing approach for BWN, including the precision self-adaptive computing approach based on SNR prediction of background noise and post-detection of network output confidence, and the time-delay based addition unit to process bit-wise approximate computing. Finally, implementation results are analyzed in section V and the paper is concluded in section VI.

II. PRELIMINARIES

A. ENERGY EFFICIENT KWS SYSTEM BASED ON BWN

On the one hand, in DNNs, the numbers of multiplication and addition operations required for the neural network layers such as convolution layers and fully connected layers are almost the same, but the power consumption of multiplication operation accounts for up to 96% of the total power consumption [9]. In our previous works [10], we have proposed a DNN network for speech recognition and a DNN accelerator architecture with approximate multiplication units to process different layers of the DNN. Implemented under 28nm CMOS technology, the power consumption of this work is 53.7 mW and the energy efficiency is 3.3 TOPS/W. However, for KWS systems, the DNN structure is too complex and the 16-bit or 8-bit width of data and weights are too redundant. On the other hand, in work [7], a binary neural network (BNN) is proposed for KWS, where the bit width of data and weight is both 1 bit. In the computation of this BNN network, the multiplication operations are almost eliminated and 99% of operations are additions and bit wise operations (e.g. XOR operations). Implemented under 28nm CMOS technology, the power consumption of this work is 141 uW and the energy efficiency is 90 TOPS/W. However, limited by the low recognition accuracy of BNN, this work can only support one keyword recognition with low background noise (SNR $\geq$ 5dB). Therefore, in our previous work [10], we have tried to use a binary weight network (BWN) structure for KWS, which can support 20 keywords recognition. In this BWN network, the weight is binarized to 1 bit, while the data is maintained as 16 bits. Similar to BNN, since the weights of BWN are also binarized to 1 bit, the multiplication operations are almost eliminated in BWN. However, since the bit width of data in BWN is not binarized to 1 bit, so that more feature characteristics of the input voice can be retained. Therefore, the recognition accuracy of BWN is much higher than BNN, which can only be applied to a few and very specific scenes. In this work, we present a KWS system which can support 10 keywords recognition under different noise types (database: Babble, White, Pink) and SNRs (≥ -5dB), based on the BWN discussed in our previous work [10]. The details of our proposed KWS system architecture and the BWN network structure will be discussed in section III.

B. APPROXIMATE COMPUTING FOR DNNS

DNNs have been proven to be naturally fault-tolerant, and the calculation accuracy requirements for various application scenarios are also in large variations [10]. Therefore, we can use approximate computing units with reduced power consumption to replace the traditional standard computing units adopted in DNNs. In our previous work [10], [11] and [13], we have proposed three digital approximate multiplication unit architectures to reduce the DNN computing power consumption. The approximate multiplication units can be dynamically reconfigured and adaptive to different accuracy requirements. Comparison results show that these approximate multiplication units can efficiently reduce the power consumption by about 50% with negligible loss of recognition accuracy. In our previous work [14], we present a voice activity detection (VAD) system based on a DNN network with two hidden fully connected layers. To further reduce the power consumption of the DNN accelerator, we proposed a digital-analog mixed multiplication unit architecture with approximate computing. This approximate computing unit can contribute a significant decrease in energy consumption by 76% $\sim$ 88% compared to that with standard computing units. For DNN processing, the power consumption of multiplications is much higher than that of other operations. Therefore, in our previous work [15], we also tried to replace most multiplication operations with addition operations in the convolution layers. This approach can significantly reduce the energy consumption of multiplication operations in convolution layers for image recognition applications with low accuracy requirements. In this work, since the DNN adopted for KWS is a BWN, more than 90% of operations are additions. However, different from the BNN adopted in work [7], where the addition operations are 1-incremental additions, in the BWN adopted for KWS in this work, the addition operations are X-incremental additions ($0 \leq X \leq 2^N$, N is the bit width of the input data of each layer). To accelerate this BWN and make it energy efficient, we proposed a time-delay based addition unit architecture and its circuit implementations. The proposed time-delay based addition unit can be dynamically reconfigured to adapt to different data bit width and computing accuracy requirements. The details of this proposed time-delay based approximate addition unit will be discussed in section IV.

III. TOP ARCHITECTURE OF KWS SYSTEM

A. OVERALL SYSTEM ARCHITECTURE

The top architecture for KWS system based on MFCC and BWN is shown in Figure 1. This KWS system mainly consists of system controller MCU implemented with ARM7TDMI, a feature extraction module, a precision control module, an SNR background detection module, a BWN accelerator, SRAMs and several assistant modules for system scheduling. The precision control module includes an iterative control unit and a bit width precision control module. The on-chip SRAMs mainly consist of voice input SRAM, feature map loading memory, weight and data memory, and each is 8 Kbytes SRAM. The MCU controls the scheduling of the whole system. The voice signal is loaded and buffered in the voice input storage unit which is sampled by A/D converter according to the frame, and then the voice data is loaded to the SNR background detection module and the feature extraction module. The system calculates the bit width according to the SNR while the feature extraction module calculates a frame of 40 speech features and then load 40 MFCCs to the BWN module. The BWN module contains four reconfigurable processing elements (PEs), and the intermediate data of the network is accessed in the on-chip data storage SRAM.

This paper trains a BWN network for keyword spotting. As shown in Figure 2, the BWN network is composed by six convolution layers and three fully connected layers. During the training process, the weights of the gradient calculations during the entire forward transfer and back propagation are binarized to +1 or −1. Quantize the feature data mapping data to 16 bits, the size of each layer of convolution kernel is 3 × 3, the number of convolution kernels is 16, 16, 32, 32, 64, 64, respectively, after each convolutional layer is the max-pooling layer, the size of the max-pooling layer is 2 × 2, each convolution layer and fully connected layer follows a batch normalization Layer.

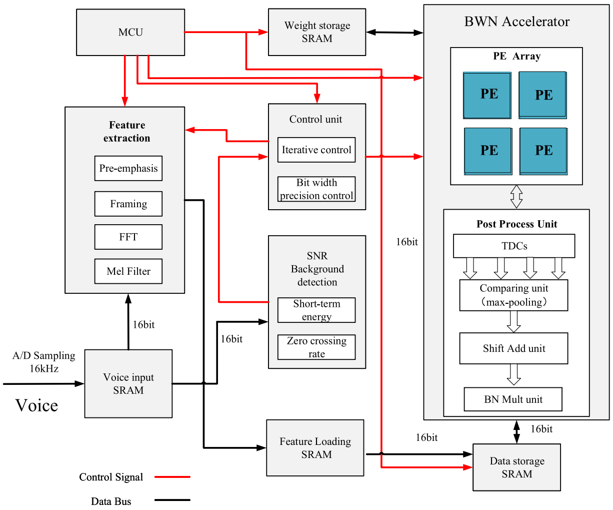


FIGURE 1. The overall architecture.

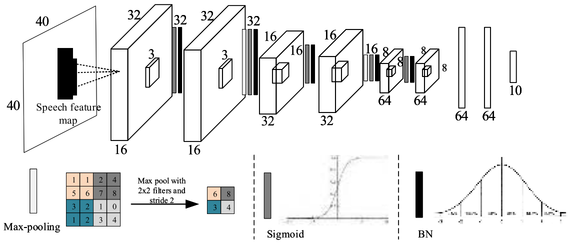


FIGURE 2. BWN network topology.

The layers used in our proposed BWN are denoted as follows: **Convolution layer:** the input filter is a 3 × 3 × 3 three-dimensional matrix. The value of each output neuron is $y = \sum_{j=1}^{3} \sum_{i=1}^{9} \omega_{ji} \cdot x_{ji} + b_j$. **Fully connected layer:** the input multi-dimensional matrix graph is expanded into one-dimensional feature vectors by row or column, then calculated with a matrix multiplication followed by a bias offset to get the value of each output neuron. The formula is:$y_j = \sum_{i=1}^{n} x_i \cdot \omega_{ji} + b_j$. **Pooling layer:** Here we choose the maximum pooling, and the size of pooling is 2 × 2. **Batch Normalization Layer:** this operation is to reduce the problem of slow convergence speed or "gradient explosion" in training. At the same time, it can speed up the training model in normal cases. The formula is:$y = \gamma \frac{x-\mu}{\sqrt{\varepsilon+\sigma^2}} + \beta$. Four parameters are represented respectively: mean, μ; variance, σ; scale, γ; offset, β. **Activation function:** without this layer, the input of each layer of the network is the linear output of the upper layer, so it is necessary to introduce a non-linear function as the activation function. Here we use sigmoid function as the activation function of output neurons in each layer. The formula is: $y = \frac{1}{e^{-x}+1}$.

For convolution operations, full connection operations, normalization operations and activation operations, their atomic operations can be regarded as various combinations of multiplication and addition. As shown in Figure 1, the PE in the BWN accelerator can be reconfigured to process multiplication or addition operation, and the interconnections between PEs can also be reconfigured by setting the data input/output registers of each PE, and therefore the BWN accelerator can process matrix multiplication and addition

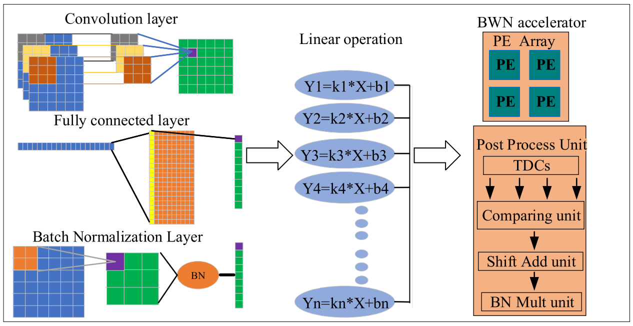


FIGURE 3. Mapping schematic diagram of BWN.

with various sizes of different layers of the adopted BWN networks for KWS. The mapping schematic diagram is shown in Figure 3. The details of PE architecture and its circuit implementation will be discussed in section IV.

B. OPTIMIZATION OF MFCC WITH APPROXIMATE COMPUTING TECHNIQUES

The feature extraction methods can be mainly classified into two types: the analog feature extraction and the digital feature extraction. Yang et al. proposed an ultra-low power voice activity detection (VAD) system based on analog feature extraction, which can reduce the power consumption to 1 μW [16]. However, due to the limitation of analog feature extraction, the proposed VAD system can only support simple speech and non-speech detection under very low background noise (SNR $\geq$ 10 dB). The digital feature extraction mainly includes the following approaches: mel-scale frequency cepstral coefficients (MFCC), linear prediction coding coefficient (LPCC) [17], perceptual linear production (PLP) [18] and rasta-plp [19]. In work [20], the advantages and disadvantages of these approaches (MFCC, LPCC, PLP, rasta-plp and other digital feature extraction approaches) are evaluated by experimental comparative analysis. Experimental and comparison results show that MFCC is a good choice when the background noise changes greatly or the SNR is low, because it has good robustness and low computational complexity. The traditional MFCC includes pre-emphasis, frame, windowing, fast Fourier transform (FFT), Mel-filtering, logarithmic operation, discrete cosine transforms (DCT), The module A shown in Figure 4 denotes the traditional MFCC.

we present a feature extraction module by optimizing the MFCC: **1.** First, we remove the gray part of the module A (traditional MFCC), including the windowing, logarithm and DCT transformation. Therefore, we can obtain module B as shown in Figure 4, which can process the feature extraction for the proposed KWS system without reducing the recognition accuracy; **2.** Second, based on the module B, we optimize the Pre-emphasis by using shift and addition operations to replace the multiplication operation; At the same time, we use rectangular bandpass filters to replace the triangular bandpass Mel filters in MEL-filtering module, and use FFT operation with no frame overlapping to reduce the computation; after these optimizations, we can obtain the module C. The detailed discussion is as follows:

The optimization of Pre-emphasis: The Pre-emphasis in Figure 4 actually passes the speech signal through a high-pass filter, expressed in the time domain as Equation 1, generally, it is between 0.9 and 1.0, usually 0.97. The purpose of pre-emphasis is to raise the high-frequency portion, flatten the spectrum of the signal, and maintain the spectrum in the same frequency-to-noise ratio in the entire frequency band from low frequency to high frequency.

$$S'_n = S_n - aS_{n-1} \tag{1}$$

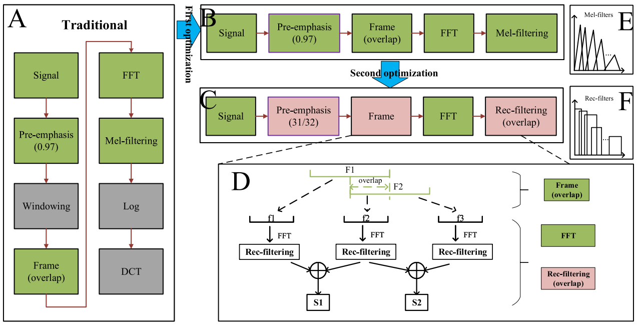


FIGURE 4. MFCC optimization for high energy efficient KWS system.

After optimization,the value of a replaces 0.97 with 31/32, as shown in Equation 2. This is to use shift and addition operations to realize the multiplication operation in the Pre-emphasis coefficient module.

$$S'_n = S_n - \frac{31}{32}S_{n-1} \tag{2}$$

The optimization of Mel filter bank: The Mel filter bank is actually a set of triangular bandpass filters of Melscale frequency, as shown by the E module in Figure 4. The frequency is converted to the Melscale frequency as shown in Equation 3:

$$Mel(f) = 1127ln(1 + \frac{f}{700}) \tag{3}$$

The specific optimization is to optimize the triangular bandpass Mel filters into rectangular bandpass filters, as shown by the F module in Figure 4. The rectangular bandpass filters are to change the amplitude of each of the triangular bandpass filters to 1 (not drawing as high as the same is to identify each filter better). For each frame that originally came out of the FFT, the data needs to pass through the triangle bandpass filters, and a series of multiply-accumulate operations are needed. Since the bandpass filters coefficients are only'1' and'0', after optimization, only addition operations are required.

The Optimization of FFT operation with no frame overlapping: The optimization of this part is by operating FFT with no frame overlapping to reduce computation as shown in the D module of Figure 4. The specific optimization details

TABLE 1. Explanation of parameters.

Parameter	Meaning
N_{level}	Precision level (bit width of data)
N_{maxout}	SNR
N_{Thred3}	the lower threshold value of SNR, when $N_{maxout} < N_{Thred3}$, set bit width as 16 bit
N_{Thred1}	the higher threshold value of SNR, when $N_{maxout} > N_{Thred1}$, set bit width as 4 bit

are as follows, where F1 and F2 are the adjacent two frames with overlapping in the original frame (where the overlap is the part with 1/2 overlap ratio). After optimization, there is no overlap between the two frames, and they are divided into three sub-frames f1, f2, f3, $f1 = f2 = f3 = \frac{1}{2}F1 = \frac{1}{2}F2$, and then FFT operations are performed on the sub-frames f1, f2, and f3, respectively. the result of FFT and rectangular filtering of f1 and f2 is added as the final feature output of the first original frame F1, the results of FFT and rectangular filters operations of f2 and f3 are added as the final feature output of the second original frame F2, such that the characteristics of all original frames are derived. This can reduce the computation by at least half by operating FFT with no frame overlapping.

IV. ENERGY-EFFICIENT APPROXIMATE COMPUTING FOR BWN

A. PRECISION CONTROL BASED ON SNR GRADING AND NETWORK OUTPUT CONFIDENCE POST-DETECTION

The KWS system with different background noise (e.g. noise types, SNRs) requires different DNN computing accuracies. For voice recognition with low background noise (e.g. SNR $\geq$ 10dB), DNN accelerator using high precision calculations with large bit width will waste computing energy; For complex noise type (e.g. the white noise) or high background noise (e.g. SNR $\leq$ 0dB), DNN accelerator using low precision calculations with low bit width may not be able to effectively process the keywords recognition. To solve this problem, we proposed a precision self-adaptive computing approach based on SNR prediction of background noise and post-detection of network output confidence.

As shown in Figure 5, the precision control system can dynamically set the bit width of data as 16 bits, 8 bits or 4 bits according to the output value of the SNR detection module and the threshold value of SNR. Table 1 denotes the parameters in Figure 5 in detail, where N_{level} represents the precision level (data bit width) used for BWN accelerator currently, and N_{maxout} represents the SNR of the current input speech signal. N_{Thred3} and N_{Thred1} respectively indicate the two thresholds value of SNR. When $N_{maxout} < N_{Thred3}$, the bit width of data will be set as 16 bits; When $N_{maxout} > N_{Thred1}$, the data bit width of data will be then set as 4 bits. When $N_{Thred1} \geq N_{maxout} \geq N_{Thred3}$, the data bit width of data will be then set as 8 bits.

The network accuracy rate(Top1 Matching Rate) under different thresholds of SNR is shown in Figure 6. Since BWN is used as the acoustic model of speech recognition, the output of the network is combined with the speech model to obtain the final result. Therefore, the probability distribution of all speech is also important, which means that the output vector should also maintain a certain precision. This work gets the deviation by computing the Euclidean Distance between the output vectors.

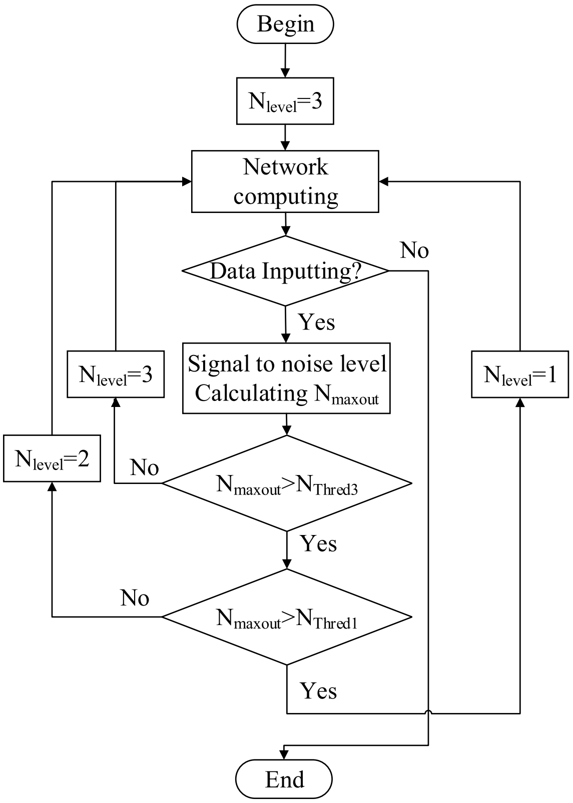


FIGURE 5. Precision control based on SNR grading and network output confidence post-detection.

According to Figure 6, some thresholds achieve lower accuracy but have relatively high power consumption. This is because the value of N_{Thred3} is too low and the value of N_{Thred1} is too high. Incorrect thresholds result in excessive calculations with bit width of 4 bits or calculations with bit width of 16 bits. Excessive calculations with bit width of 4 bits lead to a large amount of error accumulation, while too many calculations with bit width of 16 bits cause a large power consumption, thus causing a phenomenon of high power consumption and low precision. It can also be seen in the Figure 6 that by adjusting the thresholds can achieve higher precision or lower power consumption, at the optimal points (corresponding to thresholds of 0.3 and 0.8) achieve the best performance.

The output vector of the final layer will be used for classification and each value of the vector represents a classification. The output values are normalized by the network, get the sum of the vector as 1, and each value of the vector obtained represents the probability that the input voice is

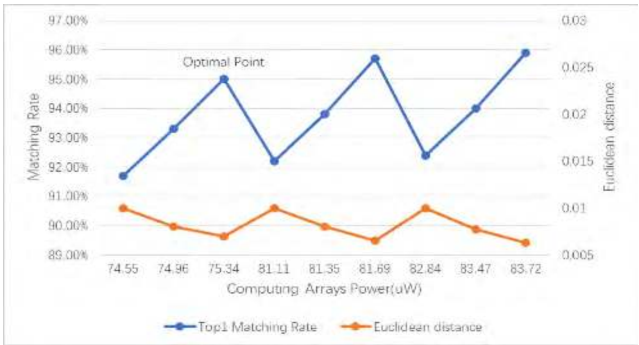


FIGURE 6. The recognition accuracy with different thresholds.

current keyword. When the recognition performance of the neural network is low, we can find that n (n≥2) output values of the output vector are relatively large. Since the sum of all output value is 1, so the maximum value of the network output is not large. Therefore, when the maximum value of BWN output vector is not high, the accuracy of the current calculation should be improved. Therefore, we can dynamically and self-adaptively set the precision level based on SNR prediction of background noise and post-detection of network output confidence.

B. IMPLEMENTATION OF PRECISION CONTROL MODULE

This module is divided into SNR detection module and bit width precision selection module. The SNR detection module detects the speech signal based on the short-time energy and short-time zero-crossing rate double threshold method to determine the complexity of the speech environment initially.

Short-time energy *ave***:** The short-term average energy of each frame is obtained by Equation 4, and the sample point n of each frame is 300. The first 10 frames are usually background noise, so we set the short-term energy threshold to the average of the first 10 frames by Equation 5.

$$ave = \frac{|X_0| + |X_1| + |X_2| + \cdots + |X_{n-1}|}{n} \tag{4}$$

$$A_{thc} = \frac{|ave_0| + |ave_1| + |ave_2| + \cdots + |ave_9|}{10} \tag{5}$$

Zero-crossing rate β**:** Different from the normal zero-crossing rate, the amplitude threshold here is not zero. The amplitude threshold is set as shown in Equation 6, where k is set to 1.3. The zero-crossing rate β used in speech noise environment complexity detection is the ratio of the absolute amplitude of each frame in 300 samples exceeding A'_{thc}. If the rate exceeds 40%, the frame is detected as normal from the perspective of zero-crossing rate.

$$A'_{thc} = k \times \frac{|ave_0| + |ave_1| + |ave_2| + \cdots + |ave_9|}{10} \tag{6}$$

In this paper, the environmental complexity γ of the SNR is set as the weighted product sum of the short-time energy and the zero-crossing rate of the normal speech signal, which is calculated as shown in Equation 7.

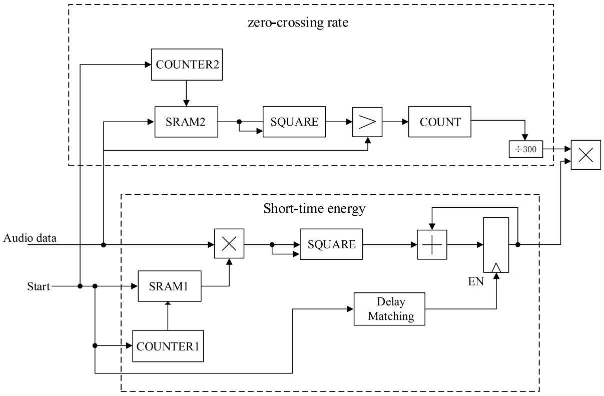


FIGURE 7. Implementation of SNR detection based on zero-crossing rate and short-time energy.

$$\gamma = ave \times (1 + \beta) \tag{7}$$

The environmental complexity γ value of one frame of the speech obtained by the SNR detection module is sent to the bit width precision selection module to classify the SNR complexity. In this work, the SNR complexity is divided into three levels, and the calculation of the network is performed using data bit width of 16 bits, 8 bits and 4 bits respectively. Its overall structure is shown in Figure 7.

C. IMPLEMENTATION OF ACTIVATION LAYER WITH APPROXIMATE COMPUTING

For the BWN adopted in our work, we use sigmoid as the activation layer. In order to reduce the circuit hardware and power consumption, we use a piecewise linear approximation method to replace the non-linear numerical operations for the activation layer. For different accuracy requirements, the appropriate number of piecewise can be selected. As mentioned in the previous section, we can obtain and evaluate the approximate SNR. Therefore, we can dynamically select and set the appropriate number of segments to calculate the sigmoid function in activation layer. When the SNR is high, the number of segments can be reduced, and thus the required numbers of calculation will be reduced. When the SNR is low, in order to improve the accuracy of calculation, the numbers of segments should be appropriately increased. In this work, we consider the linear segmentation in two cases: the number of segments N can be set as 10 or 20.

Here we refer to threshold 1 and threshold 3, taking the mean of them,the formula is: $N_{Thred2} = \frac{N_{Thred1}+N_{Thred3}}{2}$.When N$_{maxout}$ >= N$_{Thred2}$, N = 10 is chosen, and the maximum error of piecewise linear approximation is 0.0242.When N$_{maxout}$ < N$_{Thred2}$, N = 20 is chosen, the maximum error of piecewise linear approximation is 0.0075. Figure 8 shows the approximation of different numbers of segments.

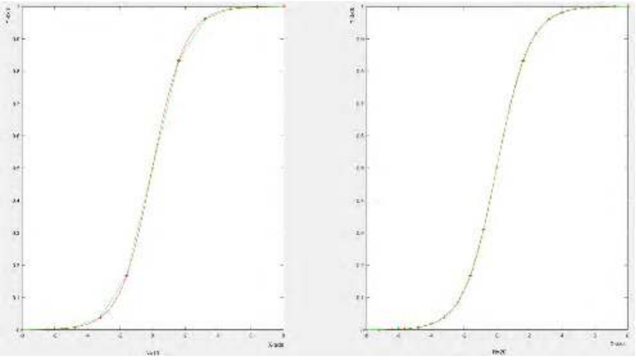


FIGURE 8. Linear approximation for activation layer computing with different numbers of segments.

Thus the calculation of non-linear functions become piecewise linear functions, as Equation 8:

$$y=\begin{cases} a_1x+b_1 & x<x_1 \\ a_2x+b_2 & x_1<x<x_2 \\ \cdots\cdots & \\ \cdots\cdots & \\ a_nx+b_n & x_{n-1}<x<x_n \end{cases} \tag{8}$$

According to the SNR in the current environment, the coefficient lookup tables corresponding to different number of segments are selected. Then, the coefficients of corresponding segments in the lookup tables are determined according to the segment intervals where the input data are located. Finally, the multiplication and addition of linear functions are realized by the two-stage pipeline method. Figure 9 shows the specific circuit logic.

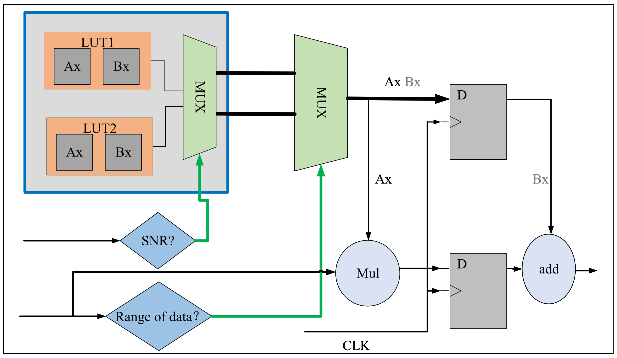


FIGURE 9. Implementation of the piecewise linear approximation for activation layer.

D. PE ARCHITECTURE WITH TIME-DELAY BASED APPROXIMATE ADDITION UNIT

As shown in Figure 10, the PE for processing BWN consists of the XOR units, time-delay based approximate addition unit with an iteration controller and a Time-to-Digital Converter (TDC). In this work, the PE can process multiplication of n×144 data (n = 1, 2, 3,···) controlled by the iteration controller at one time. 144 or n×144 data in one layer will be transmitted to the circuit unit at one time, then multiplied with the corresponding binarized weight through the XOR units. The resulting data is fed into the time-delay based addition unit for bitwise accumulation operations, then the number of'1' is summed up and converted into the delay. The data is divided into positive and negative parts before being accumulated. Both parts of the data remain the same size as the original data (fill zero in excess). Finally, the sum of the 16 corresponding bits is sent to the Post Process Unit, which conducts the accumulation, activation, normalization and pooling. Take the BWN adopted in this work as an example: for the first convolution layer, there are 16 input feature maps, and the convolution kernel is 3×3, so totally 16 × 9 = 144 data are required to be added at one time; for the second convolution layer, there are 32 input feature maps, the convolution kernel is also 3×3, so 32 × 9 = 288 data are required to be added at one time; for the third convolution layer, there are 64 input feature maps, the convolution kernel is 3×3, so 64 × 9 = 576 data are required to be added at one time. Since the delay signal can be iteratively accumulated, approximate accumulation results of various input data numbers can be obtained by configuring the PE to perform multiple iteration operations. Since there are 16 time-delay based addition units for bitwise accumulation operations in each PE, therefore, for the first convolution layer, the PE needs to perform one iteration; for the second convolution layer, the PE needs to perform two iterations; for the third convolution layer, the PE needs to perform four iterations. When the output delay signal generated by the PE is finally converted into a digital value by the TDC module, it will be then loaded to the Post Process unit for processing. When the data bit width is selected and configured as 16 bits, 8 bits and 4 bits, PE is dynamically reconfigured to be 1×16, 2×8, or 4×4 time-delay based addition units accordingly.

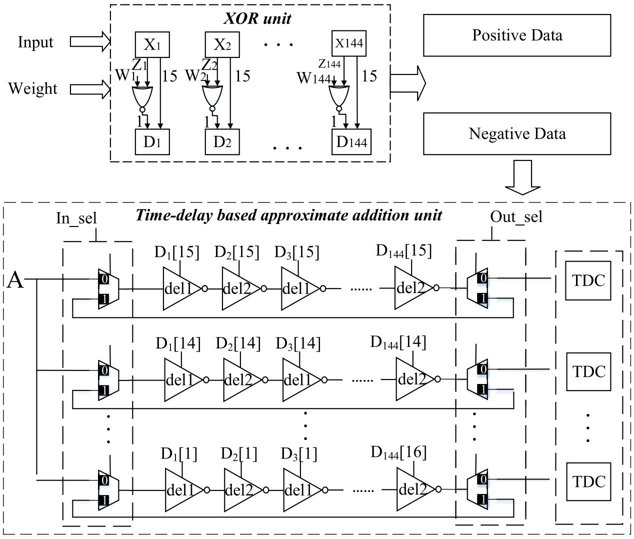


FIGURE 10. PE architecture with time-delay based addition unit.

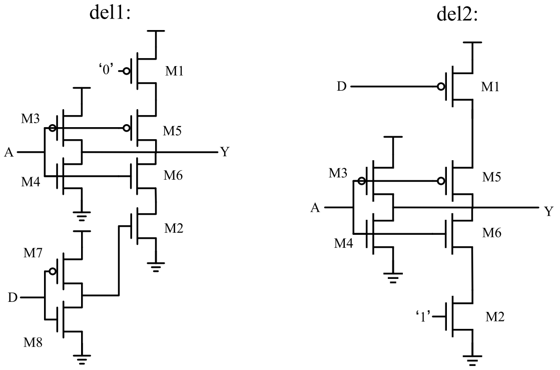


FIGURE 11. Transistor structure of the two delay blocks for time-delay based addition unit.

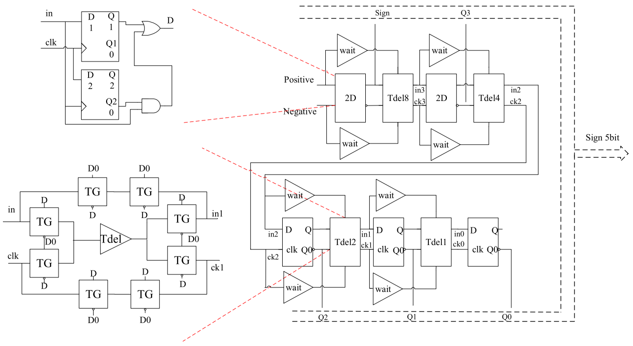


FIGURE 12. TDC architecture for time-delay based addition unit.

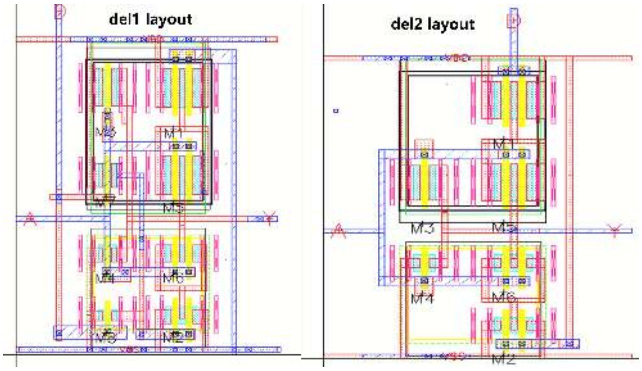


FIGURE 13. Layout of the two delay blocks: (a)del1; (b)del2.

The time-delay based addition unit consists of two kinds of delay blocks: the del1 and the del2 as shown in Figure 11. The del1 is a controllable delay block triggered by rising the edge of the clock, while the del2 is a controllable delay block triggered by the falling edge of the clock. For del1, when its control signal D is "0", M2 is turned on, and then the signal is transmitted from A to Y through M4, M6, M2. In this case, the delay is $1 \times \Delta t$. When the input signal is D is "1", M2 is turned off, then the signal is transmitted from A to Y through M4. In this case, the delay is $2 \times \Delta t$. The del2 works in the same way as del1. By changing PMOS with control D, different delays for falling edges can be obtained.

Time-delay based addition unit performs much better in energy efficiency than the conventional computation circuits. At the same time, the fluctuation of delay affected by the factors such as process corner and temperature of CMOS also brings some calculation errors. Simulation results show that under TSMC 28 nm HPC + process, the error of quantization results of the delay chain can reach 9.6%. In this work, a TDC is implemented to process the delay signal. As shown in Figure 12, the n-bit TDC based on a binary-search algorithm is composed only of n FFs and n delay blocks with different depth. It means that n-bit TDC only needs to sample the delay signal n times, which can reduce area and power consumption. To further reduce the PVT impact, the delay block of the TDC module is designed with the same delay chain as the calculation block (e.g., Tdel1, Tdel2, Tdel4 and Tdel8, as shown in Figure 12). Unlike traditional analog circuit designs, our proposed time-delay based addition unit does not need complex DAC/ADC modules. The proposed two delay blocks convert the input digital signal into a delay signal with controllable width, and the proposed TDC module quantizes a delayed pulse signal into the corresponding digital signal. Therefore, the energy efficiency and area efficiency of these modules can be effectively improved.

TABLE 2. Transistor sizes for the del1 blocks.

Size (nm)	del1 module							
Transistor	M1	M2	M3	M4	M5	M6	M7	M8
Width	800	360	380	180	800	360	240	120
Length	30	30	30	30	30	30	30	30

TABLE 3. Transistor sizes for the del2 blocks.

Size (nm)	del2 module					
Transistor	M1	M2	M3	M4	M5	M6
Width	800	360	380	180	800	360
Length	30	30	30	30	30	30

The two delay blocks are customized with Cadence Virtuoso Tool using LVT transistors under TSMC 28nm process technology. the layouts of the customized two delay blocks are shown in Figure 13. The del1 and del2 use eight and six MOS transistors, respectively. The length of each MOS transistor is 30 nm, and the width of each MOS transistor is set as a specific value, so that the time delay when the digital input is 1 is twice that when the input is 0. The specific size design for each MOS transistor of del1 and del2 is shown in Table 2 and Table 3, respectively. The layout of the customized PE Array and TDCs is shown in Figure 14.

V. IMPLEMENTATION RESULTS

The prototype system as shown in Figure 1 is implemented and evaluated on TSMC 28nm HPC+ process technology. The PE Array and TDCs of BWN accelerator is customized with Cadence Virtuoso Tool using LVT transistors, and the other digital modules are described with Verilog HDL language and synthesized by Synopsys Design Compiler (DC).

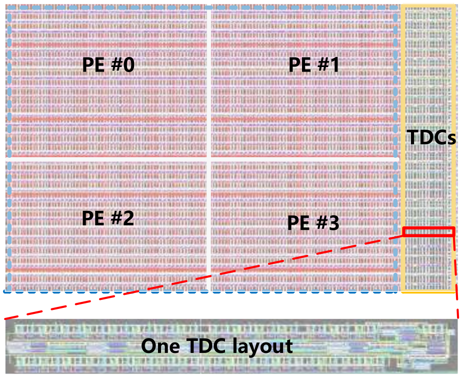


FIGURE 14. Layout of PE array and TDCs.

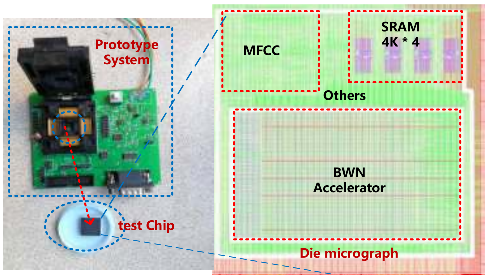


FIGURE 15. Prototype system, test chip and the die layout.

The Prototype system, the test chip and the die layout is shown in Figure 15. In the layout of PE Array and TDCs, there are 256×30 digital input pins and 8×30 digital output pins; The PE and TDC power supply connections are respectively extracted with two separate metal layers to prevent the problem of excessive local current. As shown in Figure 15, the prototype system is composed of the following components: the BWN accelerator macro, the optimized MFCC module, the four SRAM blocks and some other control modules. The area of BWN accelerator macro is 0.4×0.7 mm^2 (without memory), and the whole prototype system is 0.94 mm^2. The SRAM blocks are functional with 0.8V. The BWN Accelerator and other modules are functional with the logic supply voltage 0.6V, and the working frequency is 2.6MHz. The timing and power consumption are evaluated with Synopsys HSIM at 25°C TT corner.

We used Google's Speech Commands [21] as our training and evaluating databases. The chosen keywords are "Yes", "No", "Up", "Down", "Left", "Right", "On", "Off", "Stop", "Go", along with "silence" and "unknown". Table 4 shows the total average recognition accuracy of the prototype system presented in this paper under different data bit width (the weight bit width is 1bit). The background noise type includes white, babble and pink, the SNR includes −5db, 0dB, 5dB, 10dB, 15dB, 20dB and clear (without noise). As shown in Table 4, when the data bit width is reconfigured as 4 bits, the recognition accuracy is very low, which is only 81.6%; When the data bit width is reconfigured as 8 bits, the recognition accuracy is obviously improved, which is 89.7%; When the data bit width is reconfigured as 16 bits, the recognition accuracy can achieve 93.9%; Based on SNR prediction of background noise and post-detection of network output confidence, our proposed precision self-adaptive approach can dynamically reconfigure the data bit width as 4/8/16 bits, and achieve the recognition accuracy of 93.6%. Because our approach can dynamically set the appropriate data bit width according to the change of background noise, it can reduce the power consumption while maintaining a high recognition accuracy. Compared with the case of data bit width of 16 bits, the recognition accuracy of this work decreased by only 0.3%.

TABLE 4. KWS recognition accuracy with different data bit width.

KWS recognition accuracy (%)			
Low precision data: 4bits	Medium precision data: 8bits	High precision data: 16bits	Proposed self-adaptive precision data: 4/8/16bits
83.6	89.7	93.9	93.6

TABLE 5. Data bit width settings and recognition accuracies under different noise types and SNRs.

Noise type	SNR (dB)	Data bit width (bits)	Recognition accuracy (%)
Babble	-5	8	91.7
	0	8	92.4
	10	4	92.9
White	-5	16	89.9
	0	16	90.3
	10	8	92.1
Pink	-5	8	91.8
	0	4	92.7
	10	4	93.2

Table 5 shows the bit width chosen and setting for different noise types/SNRs and the recognition accuracy of each case, with our proposed precision self-adaptive approach. As shown in Table 5, for all the noise types and SNRs, our KWS system can achieve a high recognition accuracy (≥ 89.9%). For speech recognition, among all noise types, white noise has a full-band influence on the original input speech information, which can cause great damage to the features of the input speech spectrograms. Therefore, for white noise, when SNR is ≤ 0dB (0dB and −5dB as shown in Table 5), the prototype system will automatically use the data bit width of 16 bits, with the proposed precision self-adaptive approach; When SNR is changed to 10dB (as shown in Table 5), the prototype system will automatically change the data bit width from 16 bits to 8 bits. For babble noise, when SNR is ≤ 0dB (0dB and −5dB as shown in Table 5), the prototype system will use the data bit width of 8 bits; When SNR is changed to 10dB, the data bit width will be automatically changed to 4 bits. For pink noise, when SNR is −5dB, the prototype system will use the data bit width of 8 bits; When SNR increases to be 0dB, the data bit width will be automatically changed to 4 bits. For pink noise, the MFCC can effectively separate and eliminate the influence of noise in the input voice, so only a few data bits are needed to retain the required feature information for KWS.

The summary of power consumption evaluation with different data bit width is shown in Table 6. The power is 77.8 μW (for data bit width of 4 bits) ~ 115.9 μW (for data bit width of 16 bits) with different data bit width settings, while the throughput is 12.68 GOPS. It can be seen that the power consumption of BWN accelerator only accounts for 25% ~ 30% of the total system power consumption, which is far less than that in traditional designs. This is because the BWN accelerator in this paper uses the digital-analog mixed computing architecture based on the proposed time-delay based addition units, which is much more energy efficient than the architectures using traditional digital addition units. For our prototype system, the system power consumption is also different when the data bit width are set to different bit width (4 bits, 8 bits and 16 bits for different noise types and SNRs). When the BWN accelerator uses the data bit width of 16 bits, the total system power consumption is 115.9 μW, among which, the power consumption of the BWN accelerator and the memory access are 35.2 μW and 35.7 μW, respectively. The sum of the two (the BWN accelerator and the memory access) accounts for more than 60% of the total system power consumption. When the data bit width for BWN are 8 bits and 4 bits, the total system power consumption is reduced to 92.3 μW and 77.8 μW. This is because with the decrease of the data bit width of BWN accelerator, the power consumption of BWN computing and memory access will decrease accordingly. As shown in table 5, When the data bit width are 4 bits, the power consumption of the BWN accelerator and the memory access is 19.1 μW and 13.7 μW, respectively. The sum of the two accounts for about 40% of the total system power consumption with the data bit width of 4 bits, and is much less than that with data bit width of 16 bits which is more than 60%.

TABLE 6. Power consumption of prototype system with different data bit width.

Data bit width	Power consumption (μW) *(Percent of the total)*				
	MFCC	BWN accelerator	Memory access	Others	Total
16	36.8 *(31.75%)*	35.2 *(30.37%)*	35.7 *(30.80%)*	8.2 *(7.08%)*	115.9 *(100%)*
8	36.8 *(39.87%)*	25.2 *(27.30%)*	22.1 *(23.94%)*	8.2 *(8.89%)*	92.3 *(100%)*
4	36.8 *(47.30%)*	19.1 *(24.55%)*	13.7 *(17.61%)*	8.2 *(10.54%)*	77.8 *(100%)*

Comparisons with other state-of-the-art KWS architectures based on DNNs are shown in Table 7. In work [3] and work [4], the DNNs adopted for KWS only contain fully connected (FC) layers. In work [7] and our work, the DNNs adopted for KWS contain both FC layers and convolution neural network (CNN) layers. The CNN layers can effectively improve the recognition accuracy of KWS under low bit width of weight and data. In work [7], the proposed architecture is customized for a binarized neural network (BNN) where the bit width of data and weight are both 1 bit. To further reduce the energy consumption of the addition units, a digital approximate addition architecture is proposed in work [7]. Benefiting from the BNN network and the proposed approximate addition architecture, the power consumption of work [7] can be reduced to 141 μW and the energy efficiency of work can achieve 90 TOPS/W. In the work [3] and work [4], the bit width of data/weight are multiple bits, therefore the power consumption of the DNN accelerators mainly comes from the multiplication operations. In the computation of BNN adopted in work [7], 99% of operations are additions and the multiplication operations are almost eliminated, therefore the power consumption of the DNN accelerator can be significantly reduced, which is only 1.26% and 1.81% of the work [3] and work [4] respectively. However, since the data and weights in BNN are binary, the applicability and robustness of the BNN based KWS system are limited. In work [7], the KWS system can only support one keyword recognition under low background noise (SNR $\geq$ 5 dB). In our work, we use the BWN for KWS, where the weight is binary (1 bit) while the data is multiple bits (4/8/16 bits for different low background noise). Compared to the work [7] using BNN, our work based on BWN can support 10 keywords recognition, while most multiplication operations for BWN can also be eliminated because the weight is binary. Different from the BNN, where the addition operations are 1-incremental additions, in the BWN, the addition operations are X-incremental additions ($0 \leq X \leq 2^N$, N is the 4/8/16 bits). Compared with the work [7], the addition operations in our work will cost more hardware resources and power consumption. Therefore,

TABLE 7. Comparisons with other KWS architectures.

	ISSCC'17[4]	JSPS'16[3]	VLSI'18[7]	This work
Technology (nm)	65	40	28	28
Area (mm^2)	9.61	12	1.29	0.94
Frequency (MHz)	10.2	50	2.5	2.6
Latency (ms)	6.5	10	25	20
Voltage (V)	0.6	0.6	0.57	0.6
DNN Structure	FC	FC	CNN+FC	CNN+FC
Bit Width (Weight)	16	6	1	1
Bit Width (Data)	16	16	1	4/8/16
Computing Circuits	Standard Computing		Approximate Computing (digital)	Approximate Computing (digital-analog mixed)
Numbers of Keywords	NA	10	1	10
Power	7.78mW	11.2mW	141μW	77.8~115.9μW
Power Efficiency (TOPS/W)	NA	NA	90	109~163

we proposed the digital-analog mixed computing using time-delay based addition unit architecture and the precision self-adaptive computing using reconfigurable data bit width. Comparing with work [7], this work can achieve up to 1.8× better in energy efficiency and support 10 keywords recognition, since the digital-analog mixed computing architecture is much more energy efficient than the digital approximate computing architecture. Comparing with work [3] and work [4], this work can significantly reduce the power consumption and improve the energy efficiency.

VI. CONCLUSIONS

This paper proposed an energy-efficient reconfigurable accelerator for keyword spotting (EERA-KWS) based on binary weight network (BWN) using precision self-adaptive approximate computing. To accelerate the keyword spotting system and make it energy efficient, we presented a precision control method based on SNR prediction of background noise and post-detection of network output confidence. With this precision control method, the data path of the BWN accelerator can be dynamically and adaptively reconfigured as 4, 8 or 16 bits, and the activation layers of BWN can be computed by self-adaptive linear piecewise calculation, for various scenarios with different noise types and SNRs. Besides, we also propose a time-delay based addition unit architecture with digital-analog mixed approximate computing to further improve the BWN computing energy efficiency. This approximate computing unit can contribute a significant decrease in energy consumption compared to that with standard computing units. Implemented under TSMC 28nm technology, our work can achieve 163 TOPS/W in energy efficiency, which is over 1.8× better than the state-of-the-art architecture.

REFERENCES

[1] M. L. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Vancouver, BC, Canada, May 2013, pp. 7398–7402. doi: 10.1109/ICASSP.2013.6639100.

[2] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Trans. Audio, Speech, Language Process.*, vol. 20, no. 1, pp. 30–42, Jan. 2012. doi: 10.1109/TASL.2011.2134090.

[3] M. Shah, S. Arunachalam, J. Wang, D. Blaauw, D. Sylvester, H.-S. Kim, J.-S. Seo, and C. Chakrabarti, "A fixed-point neural network architecture for speech applications on resource constrained hardware," *J. Signal Process. Syst.*, vol. 90, no. 5, pp. 727–741, 2018. doi: 10.1007/s11265-016-1202-x.

[4] M. Price, J. Glass, and A. P. Chandrakasan, "A scalable speech recognizer with deep-neural-network acoustic models and voice-activated power gating," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, San Francisco, CA, USA, Feb. 2017, pp. 244–245. doi: 10.1109/ISSCC.2017.7870352.

[5] S. Bang, J. Wang, Z. Li, C. Gao, Y. Kim, Q. Dong, Y.-P. Chen, L. Fick, X. Sun, R. Dreslinski, T. Mudge, H. S. Kim, D. Blaauw, and D. Sylvester, "A 288 μW programmable deep-learning processor with 270 kb on-chip weight storage using non-uniform memory hierarchy for mobile intelligence," in *IEEE Int. Solid-state Circuits Conf. (ISSCC) Dig. Tech. Papers*, San Francisco, CA, USA, Feb. 2017, pp. 250–251. doi: 10.1109/ISSCC.2017.7870355.

[6] S. Yin, P. Ouyang, S. Tang, F. Tu, X. Li, S. Zheng, T. Lu, J. Gu, L. Liu, and S. Wei, "A high energy efficient reconfigurable hybrid neural network processor for deep learning applications," *IEEE J. Solid-State Circuits*, vol. 53, no. 4, pp. 968–982, Apr. 2018. doi: 10.1109/JSSC.2017.2778281.

[7] S. Yin, P. Ouyang, S. Zheng, D. Song, X. Li, L. Liu, and S. Wei, "A 141 μW, 2.46 pJ/neuron binarized convolutional neural network based self-learning speech recognition processor in 28 nm CMOS," in *Proc. IEEE Symp. VLSI Circuits*, Honolulu, HI, USA, Jun. 2018, pp. 139–140. doi: 10.1109/VLSIC.2018.8502309.

[8] K. M. Badami, S. Lauwereins, W. Meert, and M. Verhelst, "A 90 nm CMOS, 6 μW power-proportional acoustic sensing frontend for voice activity detection," *IEEE J. Solid-State Circuits*, vol. 51, no. 1, pp. 291–302, Jan. 2015. doi: 10.1109/JSSC.2015.2487276.

[9] M. Horowitz, "Computing's energy problem (and what we can do about it)," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2014, pp. 1–10. doi: 10.1109/ISSCC.2014.6757323.

[10] B. Liu, H. Qin, Y. Gong, W. Ge, M. Xia, and L. Shi, "EERA-ASR: An energy-efficient reconfigurable architecture for automatic speech recognition with hybrid DNN and approximate computing," *IEEE Access*, vol. 6, pp. 52227–52237, Sep. 2018. doi: 10.1109/ACCESS.2018.2870273.

[11] M. Pietras, "Error analysis in the hardware neural networks applications using reduced floating-point numbers representation," in *Proc. AIP Conf.*, 2015, vol. 1648, no. 1, pp. 239–255. doi: 10.1063/1.4912881.

[12] B. Liu, W. Dong, T. Xu, Y. Gong, W. Ge, J. Yang, and L. Shi, "E-ERA: An energy-efficient reconfigurable architecture for RNNs using dynamically adaptive approximate computing," *IEICE Electron. Express*, vol. 14, no. 15, pp. 1–11, 2017. doi: 10.1587/elex.14.20170637.

[13] Z. Wang, M. Xia, B. Liu, X. Ruan, Y. Gong, J. Yang, W. Ge, and J. Yang, "EERA-DNN: An energy-efficient reconfigurable architecture for DNNs with hybrid bit-width and logarithmic multiplier," *IEICE Electron. Express*, vol. 15, no. 8, pp. 1–10, 2018. doi: 10.1587/elex.15.20180212.

[14] B. Liu, Z. Wang, S. Guo, H. Yu, Y. Gong, J. Yang, and L. Shi, "An energy-efficient voice activity detector using deep neural networks and approximate computing," *Microelectron. J.*, vol. 87, pp. 12–21, Mar. 2019. doi: 10.1016/j.mejo.2019.03.009.

[15] Y. Gong, B. Liu, W. Ge, and L. Shi, "ARA: Cross-Layer approximate computing framework based reconfigurable architecture for CNNs," *Microelectron. J.*, vol. 87, pp. 33–44, May 2019. doi: 10.1016/j.mejo.2019.03.011.

[16] M. Yang, C. Yeh, and Y. Zhou, "A 1 μW voice activity detector using analog feature extraction and digital deep neural network," in *IEEE Int. Solid-State Circuits Conf. (ISSCC)Dig. Tech. Papers*, San Francisco, CA, USA, Feb. 2018, pp. 346–348. doi: 10.1109/ISSCC.2018.8310326.

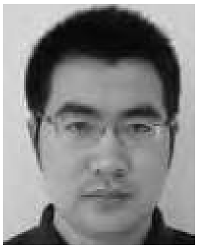
[17] D. Namrata, "Feature extraction methods LPC, PLP and MFCC in speech recognition," *Ijaret*, vol. 1, no. 6, pp. 1–4, 2013.

[18] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *J. Acoust. Soc. Amer.*, vol. 87, no. 4, pp. 1738–1752, 1990. doi: 10.1121/1.399423.

[19] H. Hermansky, N. Morgan, A. Bayya, and P. Kohn, "The challenge of inverse-E: The RASTA-PLP method," in *Proc. ACSSC*, Nov. 1991, pp. 800–804. doi: 10.1109/ACSSC.1991.186557.

[20] Z. K. Veton and A. E. Hussien, "Robust speech recognition system using conventional and hybrid features of MFCC, LPCC, PLP, RASTA-PLP and hidden Markov model classifier in noisy conditions," *J. Comput. Chem. Commun.*, vol. 3, no. 6, pp. 1–9, 2015. doi: 10.4236/jcc.2015.36001.

[21] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," 2018, *arXiv:1804.03209*. [Online]. Available: https://arxiv.org/abs/1804.03209



BO LIU (M'19) was born in Taizhou, Jiangsu, China, in 1984. He received the B.S. and Ph.D. degrees in electronic science and engineering from Southeast University, in 2006 and 2013, respectively.

He is currently a Lecturer with the National ASIC system Engineering Research Center, Southeast University. He has coauthored more than 20 academic papers and holds 30 patents. His research interests include chip architecture design, reconfigurable computing, and related VLSI designs.

ZHEN WANG received the B.S. degree in radio and the M.S. and Ph.D. degrees in electronic science and engineering from Southeast University, in 2002, 2005, and 2018, respectively. He is currently with Nanjing Prochip Electronics Technology Co., Ltd.

His research interests include calculation in memory, low power circuit, and the AI voice circuit.

HU FAN received the B.S. degree in electronic science and technology from the Henan University of Technology, Zhengzhou, China, in 2017. He is currently pursuing the M.S. degree in integrated circuit engineering from Southeast University, Nanjing, China.

His current research interests include approximate computing and neural network accelerator circuit designs.

JING YANG received the B.S. degree in electronic science and technology from the Nanjing University of Posts and Telecommunications, in 2017. She is currently pursuing the M.S degree with Southeast University, Nanjing, China.

Her current research interests include low power digital IC designs and deep learning hardware implementations.

BO LIU received the B.S. degree in microelectronics from Jilin University, Changchun, China, in 2016. He is currently pursuing the M.S. degree in electronic science and engineering with Southeast University, Nanjing, China.

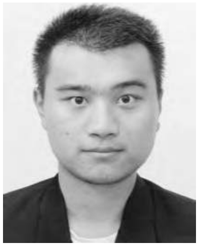
His current research interests include speech recognition and low voltage circuits.

WENTAO ZHU received the B.S. degree in electronic science and technology from the Chongqing University of Technology, Chongqing, China, in 2017. He is currently pursuing the M.S. degree in integrated circuit engineering from Southeast University, Nanjing, China.

His current research interests include speech recognition and low voltage circuits.

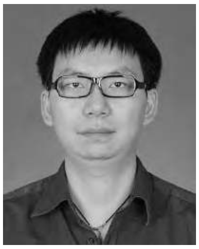
LEPENG HUANG received the B.S. degree from Soochow University, Suzhou, China, in 2017. He is currently pursuing the M.S. degree with the School of Electronic Engineering, Southeast University, Nanjing, China.

His current research interests include digital application specific integrated circuit design and neural network chip.



YU GONG received the B.S. degree in mathematics from Southeast University, in 2013, and the M.S. degree in integrated circuits from Southeast University, in 2016, where he is currently pursuing the Ph.D. degree in electronic science and engineering.

His research interests include approximate computing, reconfigurable computing, deep learning accelerator, and related VLSI design.



WEI GE received the B.S. and Ph.D. degrees in electronic science and engineering from Southeast University, in 2006 and 2015, respectively.

He is currently an Assistant Researcher with the Electrical Engineering Department, Southeast University. His research interests include SoC design technology, reconfigurable computing, and related VLSI design.

LONGXING SHI (SM'06) received the B.S., M.S., and Ph.D. degrees from Southeast University, Nanjing, China, in 1984, 1987, and 1992, respectively.

From 1992 to 2000, he was an Associate Professor with the School of Electronic Science and Engineering. Since 2001, he has been a Professor and the Dean of the National ASIC System Engineering Research Center. He has authored one book and more than 130 articles. His current research interest includes ultra-low-power IC design.