




Received January 14, 2022, accepted February 2, 2022, date of publication February 9, 2022, date of current version February 17, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3150354

Background Noise Adaptive Energy-Efficient Keywords Recognition Processor With Reusable DNN and Reconfigurable Architecture

GUOQIANG HE¹, XIAOLING DING¹², (Graduate Student Member, IEEE),
MINGHAO ZHOU¹, BO LIU¹², (Member, IEEE), AND LI LI¹¹, (Member, IEEE)

¹School of Electronic Science and Engineering, Nanjing University, Nanjing 210008, China

²National ASIC System Engineering Center, Southeast University, Nanjing 210096, China

Corresponding author: Li Li (lili@nju.edu.cn)

This work was supported by the National Key Research and Development Program of China under Grant 2018YFB2290202.

ABSTRACT This paper proposes a background noise adaptive energy-efficient keywords recognition processor with Reusable DNN (RDNN) and reconfigurable architecture. To reduce power consumption while maintaining the recognition accuracy of different background noises, the SNR prediction module determines whether the computing mode is low power consumption mode (LPM) or high performance mode (HPM). In LPM, DNN-shift (shift-based deep neural network) is used to achieve high recognition accuracy in a low background noise environment; in HPM, DNN-8bit (8bit weighted deep neural network) is used to achieve low power consumption in a high background noise environment. And the two modes share most of the hardware, and approximate computing is introduced to further reduce power consumption. Evaluated under 22nm process technology, this work can support up to 10 keywords recognition with the power consumption of 11.2 μ W for high background noise and 7.3 μ W for low background noise.

INDEX TERMS Keywords recognition, SNR prediction module, background noise adaptive, approximate computing.

I. INTRODUCTION

Keywords recognition [1]–[3] refers to the targeted recognition of some specific words in certain scenarios when the user performs intelligent voice interaction. The user can achieve the purpose of detecting whether the voice contains the keyword by customizing the keyword and the confidence level. With the rapid development of wearable devices, robots and smart homes, keyword recognition has received more and more attention, research and applications. The main indicators of keyword recognition include: power consumption, memory and recognition accuracy. On this basis, many different methods have been proposed and used to detect specific vocabulary in speech. In the past, a common solution was to apply Large-Vocabulary Continuous Speech Recognition (LVCSR) system [4] to decode the audio signal and then search for keywords in the resulting grid or network. However, these methods often require high

computing resources during decoding, and also cause system delays, which are not conducive to the portability of portable devices. In order to occupy only a small amount of memory and power consumption, as well as less calculation and delay on portable devices, small keyword recognition systems have received more and more attention.

At present, the neural network has become the mainstream keywords recognition algorithm [5], [6]. The keyword recognition system based on neural network is mainly composed of feature extraction module and keyword classification module. The speech feature extraction module mainly uses Mel Frequency Cepstrum Coefficient (MFCC) [7], [8], Linear Prediction Coefficients (LPC) [13] and other feature extraction methods. A large number of multiplications are common in both MFCC and LPC. The main structure of the keyword classification module is a deep neural network (DNN), convolutional neural network (CNN), long short-term memory network (LSTM), etc. In these structures, multiplication calculations also account for a large part of the power consumption. Therefore, it is necessary and meaning-

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

ful to use approximate multiplication instead of traditional standard multiplication to reduce power consumption.

Methods of using approximate multiplication to reduce power consumption include: in terms of software algorithms, fast convolution or deep separable convolution is used to reduce the number of convolutional layer multiplication operations. In terms of hardware design, an approximate multiplication scheme is adopted for specific modules, including digital logarithmic iterative multiplier, digital quantization look-up table, digital stage repair multiplier, and analog capacitor network multiplier. At the same time, there are specific error evaluations and precision control for specific approximate multiplication schemes. If the approximate multiplication scheme can be used when the overall network accuracy is unchanged or almost unchanged, the neural network based on the approximate multiplication unit will have a wider range of application scenarios and power consumption advantages in engineering.

Besides, in some previous research works, it was found that utilizing an approximation based on fuzzy-logic systems (FLSs) into the system can effectively reduce the complexity of the model while maintaining the accuracy of the system [9]–[12]. These works have great potential to obtain higher robustness and a wider noise adaptation range when applied to speech recognition systems based on complex neural network structures. In future research work, we will conduct research on the FLS-based keywords recognition, to further improve the recognition accuracy under high background noise.

The main contributions of this paper are described as follows.

1) A RDNN which consists of DNN-shift and DNN-8bit with different quantization methods is proposed. This RDNN can achieve high recognition accuracy and low power consumption under various background noise environments with a wide range of SNR from 0dB to clean.

2) A reconfigurable architecture based on the SNR prediction module is designed for the proposed RDNN to realize high recognition accuracy and low power consumption in different background noise scenarios. The SNR prediction module decides which computing mode (LPM or HPM) to use according to the speech complexity, and these two computing modes share most of the hardware structure. Achieve low power consumption (89.21% @clean) under low background noise, and high accuracy (83.4% @0dB) under high background noise.

3) An approximate calculation design for the reconfigurable architecture is proposed to further reduce power consumption. The approximate adder is used in LPM and the approximate multiplication cell cluster is used in HPM to reduce the number of transistors and power consumption with limited accuracy loss. This work achieves a power consumption of 11.2 μW at high background noise and 7.3 μW at low background noise.

To clearly introduce our work on keywords recognition processor, the rest parts of this paper are organized as

followed. Section II describes the related work of keyword recognition based on neural network. The energy-efficient reusable DNN for multiple background noises keywords recognition is represented in Section III. The reconfigurable architecture based on reusable DNN for keyword recognition processor is described in Section IV. Finally the implementation results are analyzed in Section V and this paper is concluded in Section VI.

II. RELATED WORKS

In recent years, hardware accelerators dedicated to neural networks have developed rapidly [14]. In order to achieve low power consumption and low cost while ensuring high accuracy, a lot of research has been done on the hardware of the keyword recognition system, and a quantitative neural network has been widely used.

In Yin's work [15], a binary convolutional neural network (BCNN) speech recognition processor was proposed. This model quantizes weights and data to 1bit, which greatly reduces the storage power consumption of data and weights. The minimum power consumption of a single wake-up word can reach 141 μW . In Giraldo's work [16], a deep neural network with long and short term memory layers was proposed. This accelerator works at a frequency of 380kHz~8MHz, which can reduce the power consumption to 2 μW ~7 μW . In our previous work [17], an ultra-low power always-on keyword spotting (KWS) accelerator was implemented in 22nm CMOS technology. We have proposed a convolutional neural network architecture based on a voltage domain analog switching network. However, the analog circuit needs to be bound to the process, and the leakage power consumption is high during digital-to-analog conversion. In our previous work [18], a high power-performance-area efficient background noise aware KWS processor based on an optimized binarized weight network (BWN) was proposed. Evaluated under 22nm process technology, the minimum power consumption for this work is 10.8 μW . However, the accuracy of the network is low, and the background noise that can be supported is not wide enough.

Binary neural network (BNN) [19] is a good choice for keyword recognition systems that require ultra-low power consumption. However, limited by the number of keywords and scenarios, quantized neural networks with larger bit widths are usually used, such as BWN [20], ternary weight neural network (TWN) [21]. In practical applications, the accuracy of the keyword recognition system based on BNN or BWN or TWN cannot be guaranteed and the flexibility is not enough.

III. ENERGY-EFFICIENT REUSABLE DNN FOR MULTIPLE BACKGROUND NOISES KEYWORDS RECOGNITION

Realizing the unification of software and hardware through quantification can effectively reduce the amount of calculation and data storage. The current mainstream quantization scheme is uniform quantization, including BNN, BWN, TWN, and DNN-8bit, etc. The DNN-shift [22] network is

a kind of non-uniform quantization. Because the weights trained in a high SNR environment present a normal distribution, the DNN-shift obtained by the non-uniform quantization operation can replace the multiplication with the shift without losing the weight expression ability, reducing the complexity of the multiplication operation in the network. The DNN-shift weight conversion method is specified as in Eq. (1). A is the operand and s is the exponent. When $s > 0$, A is shifted to the left, and when $s < 0$, A is shifted to the right. Eq. (2) the original weight W conversion method, where N is determined by the direction of W and S is determined by the size of W . The training scheme adopted is a DNN-shift for forwarding propagation and weight overall for backpropagation. Although the accuracy is reduced by 1%, there is no multiplication operation in the network, achieving low power consumption. And compared with the full-precision network, DNN-shift can still maintain high recognition accuracy while reducing the amount of calculation and complexity.

$$2^s A = \begin{cases} A \ll |s|, s > 0 \\ A \gg |s|, s < 0 \\ A, s = 0 \end{cases} \quad (1)$$

$$W * \vec{A} \rightarrow \left((-1)^N 2^S \right) * \vec{A} \quad (2)$$

The core of DNN-shift is non-uniform weight quantization. By quantizing the unlimited floating-point number W into a power of 2, the network can replace multiplication by shifting in the hardware implementation process to reduce power consumption. The optimization process of the training program is shown in Fig. 1. According to the traditional scheme, the weight initialization is randomly generated according to the normal distribution (scheme 0). For simple training sets, such as MNIST handwriting recognition, this initialization method is effective. But the effect is extremely poor when facing the task of keyword recognition. The reason is that when N and S are back propagated, the minor updates of N and S will be repaired when the integer is taken, which makes the network update effect extremely poor and the loss function cannot converge. The parameters N and S in the proposed scheme are obtained by the weight W (Eq. (1) and (2)). In response to these problems, this paper proposes two optimization schemes.

Scheme 1: Pre-train a set of available floating-point weights, then quantize them into N and S forms (conversion weights), and perform retraining (conversion training weights) on this basis. The results show that the recognition accuracy of floating-point weights is the highest, and the conversion training weights have dropped by 8-10% on the basis of floating-point weights. The conversion training weight decreased by 3.5%-4% on the basis of the floating-point weight. The training scheme of scheme 0 will have a plural in backpropagation. The imaginary part of the plural does not participate in the calculation, but only complicates the formula. Moreover, the DNN-shift network requires pre-training every time, which is very troublesome, and the

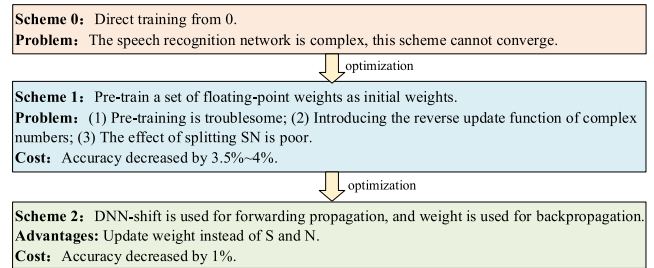


FIGURE 1. DNN-shift training scheme optimization.

TABLE 1. Recognition accuracy under different schemes.

Network structure	Recognition accuracy				
	Original weight	Conversion weight	Scheme 0	Scheme 1	Scheme 2
1Conv+3FC	79.2%	55.8%	70.3%	75.4%	78.1%
4Conv1+2FC	83.5%	78.2%	23.4%	79.5%	82.2%
4Conv2+2FC	93.23%	87.2%	N/A	88.7%	91.9%

recognition accuracy of the network decreases significantly. In response to these problems, this paper proposes a training scheme 2.

Scheme 2: There is no need to introduce a complex reverse update function and a set of pre-training weights. During the retraining process, each time it passes through the forward direction, the weights will be constrained to the power of 2 distribution, as shown in Eq. (2). In the reverse update process, the weight is taken as a whole, and the weight parameter is updated instead of the two values of S and N . Scheme 1 trains and updates N and S separately. Although it can be done theoretically, it separates the uniqueness of its weight. After trying, in a network with 12 classifications, using the optimized two scheme, the accuracy of DNN-shift only dropped by 1% compared to floating-point weights.

The comparison of DNN-shift recognition accuracy under different network structures and different training schemes is shown in Table 1. The network structures are 1Conv+3FC (30, 20, 30, 12), 4Conv1+2FC (28, 24, 16, 12, 30, 12), 4Conv2+2FC (28, 32, 24, 12, 30, 12). The size of the convolution kernel is 3×3 . In the case of the same network structure, the training effect of scheme 2 is better than scheme 0 and scheme 1. The proposed scheme2 updates weights rather than factor S and N , which eliminates complex reverse update function and a set of pre-training weights, thus greatly simplifying the training process.

To achieve high recognition accuracy, low network parameter storage and low computational power consumption in a low SNR environment, the weight is quantized to 8 bits. The traditional quantization method is a fixed-point operation, but the error distribution of this method is uneven, which leads to a decrease in recognition accuracy. In order to make the quantized weight closer to the ideal value in the training process, we adopt a bit-by-bit quantization method, as shown in Eq. (3), Eq. (4) and Eq. (5). Where W_i represents the weight parameter of the i -th layer of the neural network, k is the degree of weight quantization,

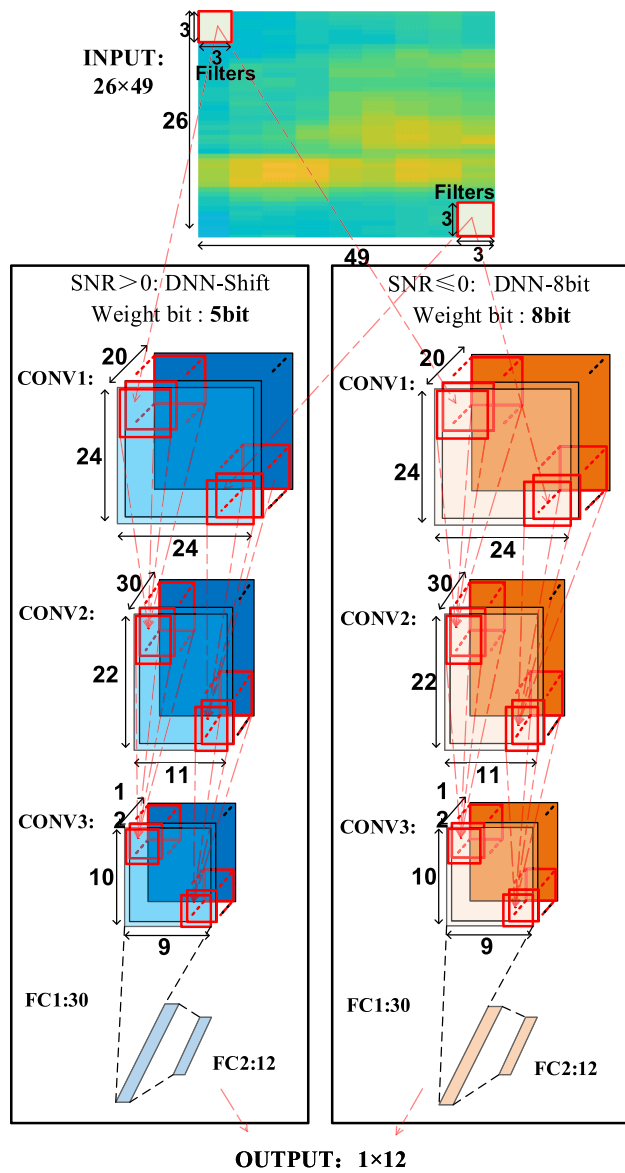


FIGURE 2. Reusable DNN architecture.

$q_k(\cdot)$ and $f(\cdot)$ represent the quantization function and the normalized compression function, respectively, and W_q is the corresponding quantization result. During quantization, the input weights are first compressed between 0 and 1. The compressed data is quantized by Eq. (3) and Eq. (5), and then the weights are converted into non-destructive fixed point numbers between -1 and 1. The high bit width quantization is performed first, and the quantized weight is saved for retraining and the quantized bit width is reduced bit-by-bit in the following training steps. This quantization method can quickly find the optimal point of network training, and perform bit-by-bit quantization at the optimal point, which not only improves the accuracy of training, but also improves the reliability of the quantized weights.

The speech dataset used in this paper comes from Google Speech Commands Dataset (GSCD), and the background

noise comes from the standard noise database Noise-92 database [24]. The 80%, 10% and 10% of the database are configured as the training dataset, validation dataset and test dataset, respectively. Evaluate the training accuracy and structural complexity of 30 network architectures in a 0dB noise environment. The final selected network architecture is shown in Fig. 2, which has a lower calculation amount and better recognition accuracy. As shown in Fig. 2, the weight width of the reusable DNN is configured as 5bit/8bit based on the SNR size, thereby realizing keywords recognition for multiple background noises.

$$q_k(W) = \frac{1}{2^k - 1} \text{round}(W * (2^k - 1)) \quad (3)$$

$$f(W_i) = \frac{W_i}{2 * \max(|W_i|)} + \frac{1}{2} \quad (4)$$

$$W_q = F^k(W_i) = 2 * q_k(f(W_i)) - 1 \quad (5)$$

IV. RECONFIGURABLE ARCHITECTURE BASED ON REUSABLE DNN FOR KEYWORD RECOGNITION PROCESSOR

A. RECONFIGURABLE ARCHITECTURE BASED ON SNR PREDICTION MODULE

The complexity of the speech signal is different in different SNR environments from the same speech, which brings challenges to keywords recognition. Therefore, this paper introduces the SNR prediction module to pre-analyze the input voice and make predictions so that the subsequent modules can be dynamically adjusted to adapt to the voice input data. The SNR prediction module classifies the input voice data by predicting the complexity of the voice and detects and evaluates the current voice environment. The SNR prediction module preliminarily predicts the speech complexity based on a reusable threshold method of short-term energy and short-term zero-crossing rate. If the SNR prediction module estimates that the SNR is high for the current voice environment, the low power mode (LPM) is adopted. If the SNR prediction module estimates that the SNR is low for the current voice environment, a high precision mode (HPM) is adopted.

In our previous work [23], to distinguish different background noise environments, the speech complexity γ which is inversely proportional to the SNR is defined. γ is based on the short-term energy α and the short-term zero-crossing rate β . By calculating the sample value of the voice data, the voice complexity γ is obtained. Fig. 3 shows the distribution of γ value under different background noises. It can be seen from the figure that the noise type has little effect. When SNR=0dB, the γ value remains around 30. Therefore, this paper uses 30 as the threshold to divide the voice noise environment into high and low. Determine the subsequent network accuracy selection through the SNR prediction module. The SNR prediction module is shown in Fig. 4, the speech signal is classified into a low SNR speech signal and a high SNR speech signal in the SNR prediction module, corresponding to HPM and LPM respectively.

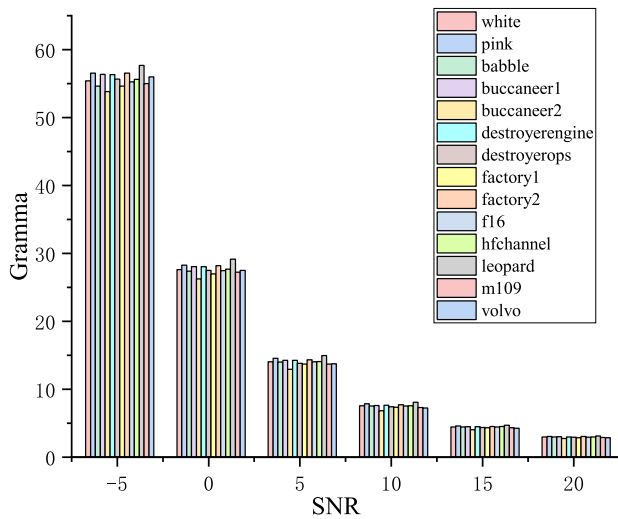


FIGURE 3. Distribution of γ value under different background noises.

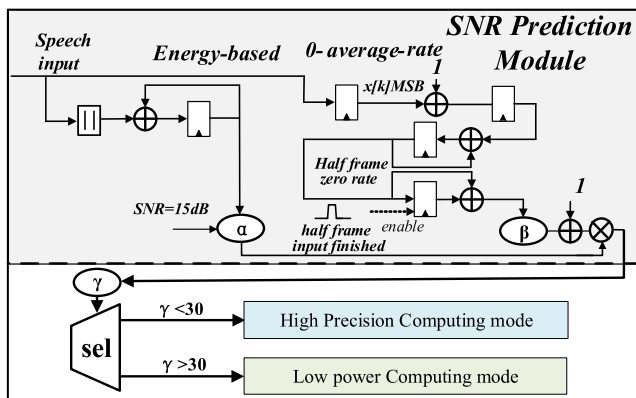


FIGURE 4. SNR prediction module.

For the system architecture, common modules including Layer Controller module, Memory module, Memory controller module and BN_RELU optimization module are designed. And a reconfigurable PE module is designed for HPM and LPM calculations. As shown in Fig. 5(a), after the speech signal is processed by MFCC, the classification result is obtained through the neural network classification module. As shown in Fig. 5(a), there are three SRAMs in the hardware for storing voice data and weights under different SNR environments. The Memory controller module is used to generate addresses and call specific data in SRAM during calculations. The layer controller uses the state machine to control the current layer of the computing network. The BN_RELU module is shown in Fig. 5(a), using a pipeline design to replace the multiplication operation with multi-bit data addition, and complete the BN_RELU operation in one beat. According to the low 8 bits in the 16bit*8bit result as invalid data, the cut-off selection at the initial stage of the multiplication is performed, and the effective bit width of 16bit is directly calculated. Fig. 5(b) shows the reconfigurable PE module, which is used for network calculations. The results obtained by the two modes share

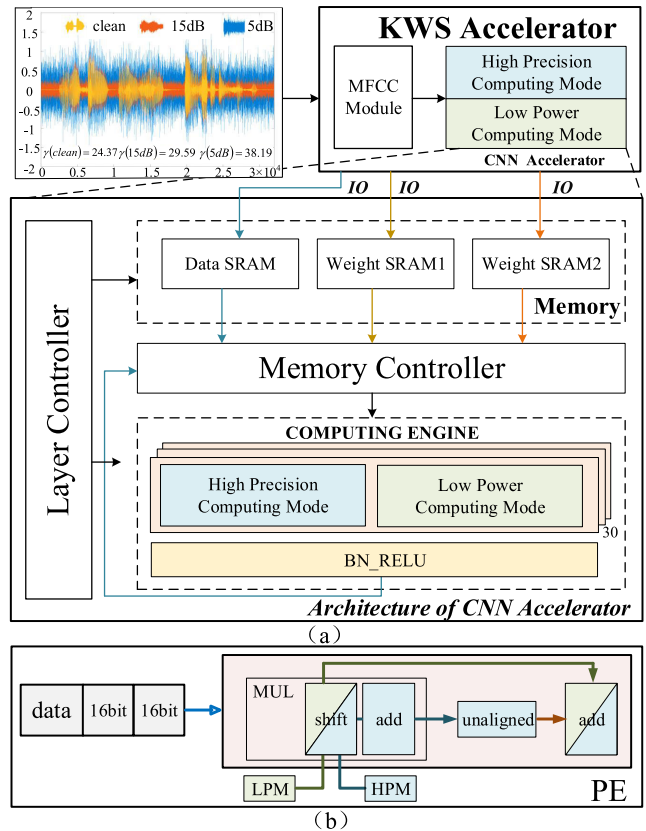


FIGURE 5. (a) CNN accelerator architecture. (b) Design of reconfigurable PE module.

a BN_RELU module, and the data after each layer of convolution operation is normalized. In LPM, the PE module only performs shifting and accumulating operations, that is, turning off the blue operation in the PE template to achieve low power consumption; in HPM, the PE module performs traditional multiplication shifting and accumulating operations, aligning and accumulating operations to achieve high precision.

The BN_RELU optimization module mainly completes the functions of the BN layer and the ReLU activation layer. In terms of hardware implementation, if there is no BN layer to constrain the data, then the network data value will increase explosively, so the BN layer is used to normalize the data. The function of the BN circuit module is to add the offset to the result of multiplying the input data D_{in} and scale to obtain the D_{O} output result. In order to realize the pipeline operation, the register R5 is added in Fig. 6(a), which is convenient for calculation. In the circuit design of the ReLU module of Fig. 6(b), the input data is judged by the highest sign bit and 0. If the input data is greater than 0, the output remains unchanged, and if the input data is less than 0, the output is set to 0. The optimized BN_RELU circuit module is shown in Fig. 6(c), and the multiplication operation is replaced with multi-bit data addition using a pipeline design, thereby completing the BN_RELU operation in one beat. This proposed BN_RELU module can greatly reduce the

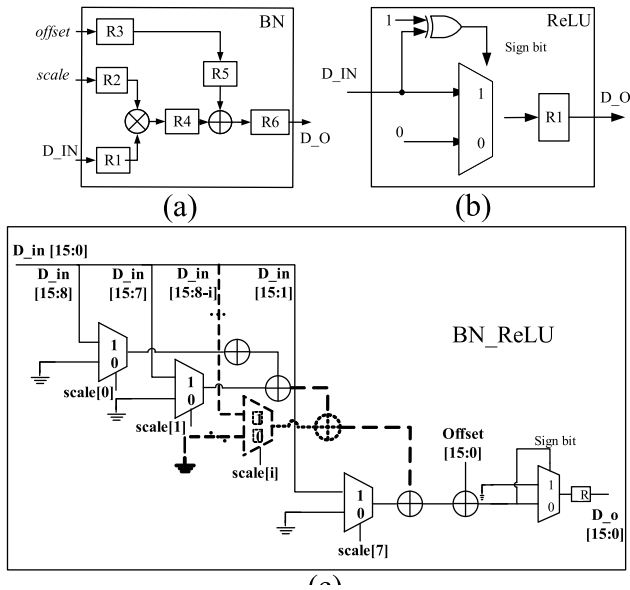


FIGURE 6. (a) BN circuit module. (b) ReLU circuit module. (c) Optimized BN_ReLU circuit module.

power consumption and achieve high throughput, that is to achieve high energy efficiency.

B. APPROXIMATE COMPUTING DESIGN FOR RECONFIGURABLE ARCHITECTURE

For different computing modes, different approximate computing methods are used. Because there are only shift and accumulation calculations in LPM, the approximate adder proposed in our previous work [18] can effectively reduce power consumption within 0.5% of the accuracy loss. In HPM, the design is a multiplier with an input bit width of 16bit*8bit and an output bit width of 16bit. Table 2 shows the amount of calculation for multiplication and addition in DNN-8bit. In this network algorithm, although the amount of calculation for multiplication is equivalent to the amount of calculation for addition because the complexity of multiplication is much higher than that of addition, the power consumption of multiplication is 6 times that of addition. In order to still achieve low power consumption in a low SNR environment, this paper optimizes the power consumption of the multiplication in DNN-8bit. Due to the fault tolerance of the neural network, the approximate multiplication unit designed in this paper can be realized based on low power consumption without affecting the recognition accuracy of the network.

In the DNN-8bit algorithm, the convolution kernel used is a fixed 3 × 3. Based on this, an approximate multiplication unit cluster design for DNN-8bit is proposed. Unlike the approximate multiplication unit, the input is no longer a single 16bit data and 8bit weight, but a 9*16bit data array and 9*8bit weight array. In the coding method, the R8ABE2 [25] coding scheme that introduces positive errors is used to offset the negative errors introduced by the carry-and-discard line. First, perform R8ABE2 encoding on 9 sets of data and

TABLE 2. Network calculation amount in DNN-8bit.

Input data	Output result	Multiplication calculation amount	Addition calculation amount
49*26	24*24*20	24*24*20*3*3	24*24*20*3*3
24*24*20	22*11*30	22*11*30*3*3*20	22*11*30*3*3*20
22*11*30	10*9*12	10*9*12*3*3*30	10*9*12*3*3*30
10*9*12	30	10*9*12*30	10*9*12*30
30	12	30*12	30*12
N/A	N/A	1734840 (Total calculation)	1734840 (Total calculation)

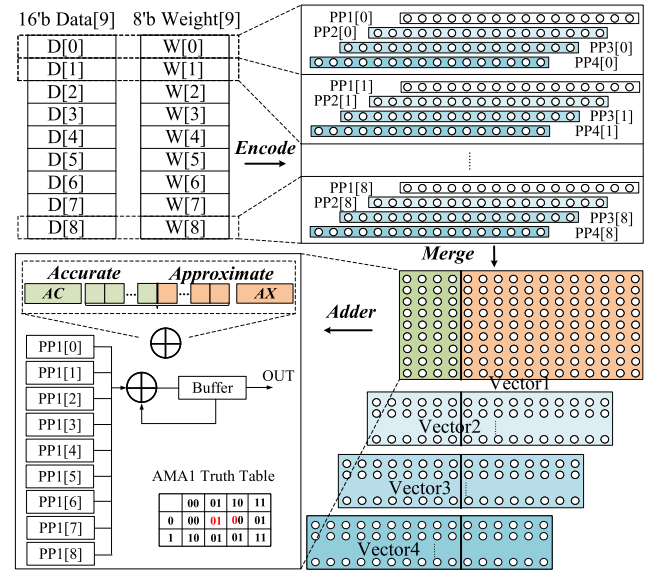


FIGURE 7. Approximate multiplying unit clusters.

weights respectively to obtain 9 partial product arrays as shown in Fig. 7. In a partial product array, 4 rows of partial products have different effects on the final result, so the 9 partial product arrays are rearranged and merged, and the partial products at the same position are merged to obtain 4 new partial products. The array is shown in Fig. 7. For each array, it is equivalent to accumulating 9*16bit numbers, and an approximate addition unit is used for each layer. Fig. 7 shows the accumulation process of PP1. For the summation of 9 partial products, the method of successive accumulation is adopted, and only one addition unit is required to complete the accumulation operation. Although it uses more cycles than the parallel addition tree design, it can be completed with only one circuit, reducing the circuit area. Moreover, the successive accumulation can prevent data overflow in the accumulation process and reduce errors by pre-expanding the bit width. Finally, perform a shift arrangement and sum the 4 partial products to obtain the sum of the products of the 9*16bit data array and the 9*8bit weight array.

As shown in Table 3, in the approximate multiplication unit cluster, an approximate adder is introduced in the lower position. The experimental comparison shows that the AMA1 approximate addition scheme only produces a 1.4% error when approximately 12 bits and the number of transistors can be reduced by 32%. The approximate multiplication unit

TABLE 3. Approximate Multiplication Unit Cluster Scheme Based on Approximate Addition.

Approximate scheme	Approximately low 8 bits	Approximately low 10 bits	Approximately low 12 bits	Approximately low 14 bits	Number of transistors
ACC	0	0	0	0	28*16
AMA1	0.00008052	0.0010754	0.01412	0.266	16*12+28*4
AMA2	0.0000189	0.0016252	0.0414	0.4141	14*12+28*4
AMA3	0.0001644	0.002642	0.0336	0.6712	11*12+28*4
AMA4	0.000146	0.0023310	0.0375	0.5931	11*12+28*4
AMA5	0.00002037	0.001955	0.026	0.4191	10*12+28*4
TGA1	0.00045714	0.001466	0.0117	0.2721	16*12+28*4
TGA2	0.00003782	0.0005921	0.0089	0.471	22*12+28*4

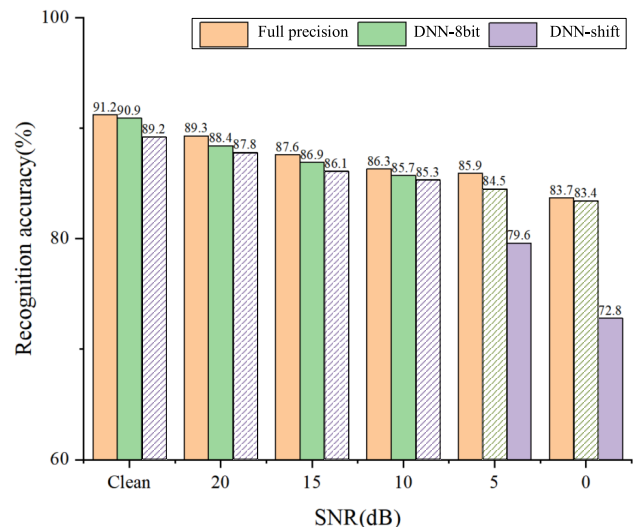
TABLE 4. Comparison with the state-of-art KWS architecture.

	VLSI [15] '18	ACCESS [17] '18	VLSI [26] '19	ISSC [27] '20	TCAS-I [18] '20	This work
Technology	28nm	22nm	65 nm	28 nm	22 nm	22 nm
Architecture	MFCC+BCNN	MFCC+CNN	MFCC+LSTM	MFCC+DSCNN	MFCC+BWN	MFCC+RDNN
Bit width (Data)	1bit	8bits	MFCC:10bits LSTM:48bits	1bit	16bits	MFCC:16bits RDNN:16bits
Bit width (Weight)	1bit	7bits	8bits	1bit	1bit	DNN-8bit:8bits DNN-shift:5bits
Frequency	2.5MHz	250KHz	250KHz	40KHz	250KHz	250KHz
Latency	0.5~25 ms	20 ms	16 ms	64 ms	16 ms	16 ms
Voltage	0.57V	0.55V	0.6V - 1.2V	0.41V	0.6V	0.6V
Layout Area	1.29mm ²	0.75mm ²	2.56mm ²	0.23mm ²	0.6mm ²	2.14mm ²
Memory	52kB	58kB	65kB	2kB	11kB	26.61kB
Numbers of Keywords	1	10+unknown&silence	10+unknown&silence	1-2	10+unknown&silence	10+unknown&silence
Power	141μW	52μW	18.3 μW	0.51 μW	10.8 μW@AC mode 15.1 μW@SC mode	7.3μW@LPM 11.2μW@HPM
Background noise support	SNR:5dB~clean	SNR:-5dB~clean	Clean	White Noise SNR:5dB~Clean	SNR:5dB~Clean	SNR:0dB~Clean
Database	TIDIGIT	GSCD	GSCD&TIDIGITS	GSCD	GSCD	GSCD
Recognition accuracy	Clean:95% 10dB:88% 5dB:85%	Clean:90.51% 10dB:89.13% 0dB:83.62% -5dB:81.12%	Clean:90.87% @GSCD	1 word: clean:98% 30dB:93.6% 5dB:90.7%	Clean:87.9% 10dB:84.4% 5dB:80.8%	Clean:89.21%(LPM) 10dB:85.35%(LPM) 0dB:85.51%(HPM)

cluster based on AMA1 is applied to the network for joint simulation and the comparison of recognition accuracy under different SNRs of the three networks is shown in Fig. 8. In a high SNR environment, the recognition accuracy of the full-precision network, DNN-8bit and DNN-shift is basically the same. In a low SNR environment, DNN-shift is no longer applicable, and the recognition accuracy of DNN-8bit is close to that of the full-precision network. We adapt both DNN-shift and DNN-8bit for low and high SNR respectively, which can achieve high recognition accuracy under high background noise and ultra-low power consumption under low background noise.

V. IMPLEMENTATION RESULTS

In order to evaluate the power consumption and recognition accuracy of the proposed architecture and circuit design method, the implemented keywords recognition system was evaluated on the TSMC 22nm ULL process technology. In a high SNR environment, the DNN-shift is used, and the power consumption is 7.3μW. In a low SNR environment, the DNN-8bit is used, and the power consumption is 11.2μW. The overall hardware layout is shown in Fig. 9, with an area of 1.845*1.160mm². Table 4 shows the comparison with the state-of-the-art keywords recognition architecture based on neural networks. The feature extraction algorithm used in the reference works and our work are MFCC. From the perspective of keywords recognition environment, compared with the results in recent years, the solution in this paper is

**FIGURE 8. Comparison of recognition accuracy under different SNRs of three networks.**

suitable for speech recognition in an environment with SNR of 0dB to 20dB, and the supported SNR range is wider, while other solutions are only suitable for SNR higher than 10dB.

Compared with the work using BCNN [15], our work can support the recognition of 10 keywords, and the supported SNR range is wider. Compared with work [17], since there is no leakage power consumption between digital-to-analog conversion, our work achieves similar accuracy while

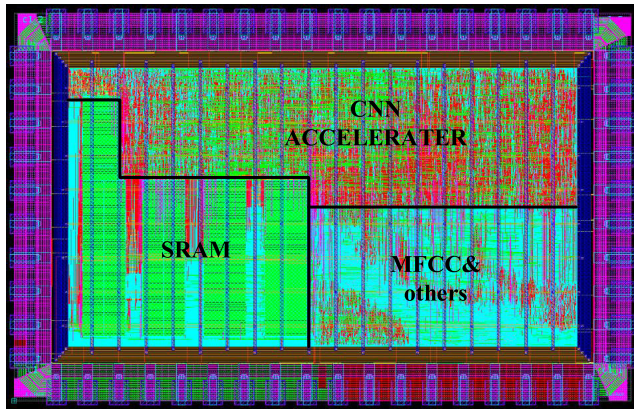


FIGURE 9. Layout of the speech keywords recognition prototype system with proposed reconfigurable architecture.

lowering power consumption. Compared with work [27], the power consumption of this paper is higher. This is because this work has completed 10 keywords recognition in a multi-SNR environment. Compared with work [26], the power consumption of this article to realize 10 keyword recognition is smaller. Compared with work [18], although the two computing modes share most of the hardware structure, there is still a certain amount of hardware redundancy. In addition, since the weights of DNN-shift and DNN-8bit need to be stored at the same time, more memory is required. But our work can achieve higher recognition accuracy with lower power consumption. The proposed keywords recognition architecture uses the DNN-shift in a clean environment to achieve a recognition accuracy of 89.21% and an accuracy of 85.3% in a 10dB environment. A network recognition accuracy of 83.4% can be achieved using DNN-8bit in a noise environment of 0dB.

VI. CONCLUSION

In this paper, we proposed a background noise adaptive energy-efficient keywords recognition processor with reusable DNN and reconfigurable architecture. In a high SNR environment, the DNN-shift is configured to LPM to achieve low power consumption; in a low SNR environment, the DNN-8bit is configured to HPM to achieve high accuracy. With this approach, the regulation processor can dynamically achieve the balance between high accuracy and low power consumption. The prototype system is implemented and evaluated under the TSMC 22nm ULL CMOS process technology. Achieve low power consumption (89.21%@clean) under low background noise, and high accuracy (83.4%@0dB) under high background noise.

REFERENCES

- [1] R. C. Rose and D. B. Paul, "A hidden Markov model based keyword recognition system," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Apr. 1990, pp. 129–132, doi: [10.1109/ICASSP.1990.115555](https://doi.org/10.1109/ICASSP.1990.115555).
- [2] 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](https://doi.org/10.4236/jcc.2015.36001).
- [3] Y. Wang, P. Getreuer, T. Hughes, R. F. Lyon, and R. A. Saurous, "Trainable frontend for robust and far-field keyword spotting," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 5670–5674, doi: [10.1109/ICASSP.2017.7953242](https://doi.org/10.1109/ICASSP.2017.7953242).
- [4] J. G. Wilpon, L. Rabiner, C.-C. Lee, and E. R. Goldman, "Automatic recognition of keywords in unconstrained speech using hidden Markov models," *IEEE Trans. Acoust., Speech Signal Process.*, vol. 38, no. 11, pp. 1870–1878, Nov. 1990, doi: [10.1109/29.103088](https://doi.org/10.1109/29.103088).
- [5] 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, May 2018, doi: [10.1007/s11265-016-1202-x](https://doi.org/10.1007/s11265-016-1202-x).
- [6] S.-G. Leem, I.-C. Yoo, and D. Yook, "Multitask learning of deep neural network-based keyword spotting for IoT devices," *IEEE Trans. Consum. Electron.*, vol. 65, no. 2, pp. 188–194, May 2019, doi: [10.1109/TCE.2019.2899067](https://doi.org/10.1109/TCE.2019.2899067).
- [7] K. Umaphathy, R. K. Rao, and S. Krishnan, "Audio signal feature extraction and classification using local discriminant bases," in *Proc. Int. Conf. Signal Process. Commun. (SPCOM)*, May 2004, pp. 457–461, doi: [10.1109/SPCOM.2004.1458501](https://doi.org/10.1109/SPCOM.2004.1458501).
- [8] K. Bhattacharai, P. W. C. Prasad, A. Alsadoon, L. Pham, and A. Elchouemi, "Experiments on the MFCC application in speaker recognition using MATLAB," in *Proc. 7th Int. Conf. Inf. Sci. Technol. (ICIST)*, Apr. 2017, pp. 32–37, doi: [10.1109/ICIST.2017.7926796](https://doi.org/10.1109/ICIST.2017.7926796).
- [9] L. Fang, S. Ding, J. H. Park, and L. Ma, "Adaptive fuzzy control for stochastic high-order nonlinear systems with output constraints," *IEEE Trans. Fuzzy Syst.*, vol. 29, no. 9, pp. 2635–2646, Sep. 2021, doi: [10.1109/TFUZZ.2020.3005350](https://doi.org/10.1109/TFUZZ.2020.3005350).
- [10] L. Fang, S. Ding, J. H. Park, and L. Ma, "Adaptive fuzzy control for nontriangular stochastic high-order nonlinear systems subject to asymmetric output constraints," *IEEE Trans. Cybern.*, early access, Jun. 29, 2020, doi: [10.1109/TCYB.2020.3000920](https://doi.org/10.1109/TCYB.2020.3000920).
- [11] S. Ding, K. Mei, and X. Yu, "Adaptive second-order sliding mode control: A Lyapunov approach," *IEEE Trans. Autom. Control*, early access, Sep. 24, 2021, doi: [10.1109/TAC.2021.3115447](https://doi.org/10.1109/TAC.2021.3115447).
- [12] K. Mei, S. Ding, and W. X. Zheng, "Fuzzy adaptive SOSM based control of a type of nonlinear systems," in *IEEE Trans. Circuits Syst. II, Exp. Briefs*, early access, Sep. 30, 2021, doi: [10.1109/TCSII.2021.3116812](https://doi.org/10.1109/TCSII.2021.3116812).
- [13] P. G. Nidhi, S. P. D. Dakshayani, S. Sushma, V. Neelima, and B. K. Priya, "Implementation of linear prediction coefficients in G.729E using VHDL for man mission applications," in *Proc. Int. Conf. Smart Syst. Inventive Technol. (ICSSIT)*, Nov. 2019, pp. 240–245, doi: [10.1109/ICSSIT46314.2019.8987855](https://doi.org/10.1109/ICSSIT46314.2019.8987855).
- [14] 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. Exp.*, vol. 14, no. 15, 2017, Art. no. 20170637, doi: [10.1587/ele.14.20170637](https://doi.org/10.1587/ele.14.20170637).
- [15] S. Yin, P. Ouyang, S. Zheng, D. Song, X. Li, L. Liu, and S. Wei, "A 141 UW, 2.46 PJ/Neuron binarized convolutional neural network based self-learning speech recognition processor in 28NM CMOS," in *Proc. IEEE Symp. VLSI Circuits*, Jun. 2018, pp. 139–140, doi: [10.1109/VLSIC.2018.8502309](https://doi.org/10.1109/VLSIC.2018.8502309).
- [16] J. S. P. Giraldo and M. Verhelst, "Laika: A 5 μ W programmable LSTM accelerator for always-on keyword spotting in 65nm CMOS," in *Proc. ESSCIRC IEEE 44th Eur. Solid State Circuits Conf. (ESSCIRC)*, Sep. 2018, pp. 166–169, doi: [10.1109/ESSCIRC.2018.8494342](https://doi.org/10.1109/ESSCIRC.2018.8494342).
- [17] B. Liu, Z. Wang, W. Zhu, Y. Sun, Z. Shen, L. Huang, Y. Li, Y. Gong, and W. Ge, "An ultra-low power always-on keyword spotting accelerator using quantized convolutional neural network and voltage-domain analog switching network-based approximate computing," *IEEE Access*, vol. 7, pp. 186456–186469, 2019, doi: [10.1109/ACCESS.2019.2960948](https://doi.org/10.1109/ACCESS.2019.2960948).
- [18] B. Liu, H. Cai, Z. Wang, Y. Sun, Z. Shen, W. Zhu, Y. Li, Y. Gong, W. Ge, J. Yang, and L. Shi, "A 22nm, 10.8 μ W/15.1 μ W dual computing modes high power-performance-area efficiency dominated background noise aware keyword-spotting processor," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 12, pp. 4733–4746, Dec. 2020, doi: [10.1109/TCSI.2020.2997913](https://doi.org/10.1109/TCSI.2020.2997913).
- [19] D. Jaeger and R. Jung, "Binary neural network," in *Encyclopedia of Computational Neuroscience*. Cham, Switzerland: Springer, 2015, p. 385, doi: [10.1007/978-1-4614-6675-8_100065](https://doi.org/10.1007/978-1-4614-6675-8_100065).
- [20] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "XNOR-Net: ImageNet classification using binary convolutional neural networks," in *Proc. Eur. Conf. Comput. Vis.*, vol. 9908, 2016, pp. 525–542, doi: [10.1007/978-3-319-46493-0_32](https://doi.org/10.1007/978-3-319-46493-0_32).

- [21] F. Li, B. Zhang, and B. Liu, "Ternary weight networks," 2016, *arXiv:1605.04711*.
- [22] M. Elhoushi, Z. Chen, F. Shafiq, Y. H. Tian, and J. Y. Li, "DeepShift: Towards multiplication-less neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2021, pp. 2359–2368.
- [23] B. Liu, Y. Li, L. Huang, H. Cai, W. Zhu, S. Guo, Y. Gong, and Z. Wang, "A background noise self-adaptive VAD using SNR prediction based precision dynamic reconfigurable approximate computing," in *Proc. Great Lakes Symp. (VLSI)*, Sep. 2020, pp. 271–275, doi: [10.1145/3386263.3407589](https://doi.org/10.1145/3386263.3407589).
- [24] A. Varga and H. J. M. Steeneken, "Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems," *Speech Commun.*, vol. 12, no. 3, pp. 247–251, Jul. 1993, doi: [10.1016/0167-6393\(93\)90095-3](https://doi.org/10.1016/0167-6393(93)90095-3).
- [25] H. Waris, C. Wang, and W. Liu, "Hybrid low radix encoding-based approximate booth multipliers," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 67, no. 12, pp. 3367–3371, Dec. 2020, doi: [10.1109/TCSII.2020.2975094](https://doi.org/10.1109/TCSII.2020.2975094).
- [26] J. S. P. Giraldo, S. Lauwereins, K. Badami, H. Van Hamme, and M. Verhelst, "18 μ W SoC for near-microphone keyword spotting and speaker verification," in *Proc. Symp. VLSI Circuits*, Jun. 2019, pp. C52–C53, doi: [10.23919/VLSIC.2019.8777994](https://doi.org/10.23919/VLSIC.2019.8777994).
- [27] W. Shan, M. Yang, J. Xu, Y. Lu, S. Zhang, T. Wang, J. Yang, L. Shi, and M. Seok, "14.1 A 510nW 0.41 V low-memory low-computation keyword-spotting chip using serial FFT-based MFCC and binarized depthwise separable convolutional neural network in 28nm CMOS," in *IEEE ISSCC Dig. Tech. Papers*, Feb. 2020, pp. 230–232, doi: [10.1109/ISSCC19947.2020.9063000](https://doi.org/10.1109/ISSCC19947.2020.9063000).



ture research of high-performance digital signal processor.

GUOQIANG HE received the B.S. degree in electronics information technology from the Wuhan University of Technology, Wuhan, China, in 2000, and the M.S. degree in electronics and communication engineering from the Nanjing University of Technology, Nanjing, China, in 2010. He is currently pursuing the Ph.D. degree in electronic science and technology with Nanjing University. His current research interests include VLSI design for digital signal processing systems and architecture research of high-performance digital signal processor.



XIAOLING DING (Graduate Student Member, IEEE) received the B.S. degree in electronic information science and technology from Henan University, Henan, China, in 2019. She is currently pursuing the M.S. degree in integrated circuit engineering with Southeast University, Nanjing, China. Her current research interests include speech recognition and low voltage circuits.



MINGHAO ZHOU received the B.S. degree in electronic science and engineering from Nanjing University, Jiangsu, China, in 2004, and the master's degree in electronic and information engineering from Shanghai Jiao Tong University, Shanghai, China, in 2007. He is currently pursuing the Ph.D. degree in electronic science and engineering with Nanjing University. His current research interests include 2.5D/3D systems integration packaging technologies and thermal dissipation technology for ASIC and 3DIC.



research was supported by the National Natural Science Foundation, the National Science and Technology Major Project, and the National Key Research and Development Program. His research interests include chip architecture design, reconfigurable computing, approximate computing, and related VLSI designs.

BO LIU (Member, IEEE) 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 an Associate Professor with the National ASIC System Engineering Research Center, Southeast University. He has authored or coauthored more than 40 scientific papers in the above research fields, and holds one U.S. patent and over 30 Chinese patents. His



a member of the Circuits and Systems for Communications (CASCOM) TC of IEEE CAS Society.

LI LI (Member, IEEE) received the B.S. and Ph.D. degrees from the Hefei University of Technology, Hefei, China, in 1996 and 2002, respectively. She is currently a Professor at the School of Electronic Science and Engineering, VLSI Design Institute, Nanjing University, Nanjing, China. Her current research interests include VLSI design for digital signal processing systems, reconfigurable computing, and multiprocessor system-on-a-chip (MPSoC) architecture design methodology. She is

...