A 1D-CRNN Inspired Reconfigurable Processor for Noise-robust Low-power Keywords Recognition

Bo Liu, Zeyu Shen, Lepeng Huang, Yu Gong, Zilong Zhang, Hao Cai
National ASIC System Engineering Research Center
Southeast University
Nanjing, China
{liubo_cnasic, zeyushen, huanglepeng, gongyu, zhangzilong, hao.cai}@seu.edu.cn

***Abstract*—A low-power high-accuracy reconfigurable processor is proposed for noise-robust keywords recognition and evaluated in 22nm technology, which is based on an optimized one-dimensional convolutional recurrent neural network (1D-CRNN). In traditional DNN-based keywords recognition system, the speech feature extraction based on traditional algorithms and the DNN based keywords classification are two independent modules. Compared to the traditional architecture, both the feature extraction and keywords classification are processed by the proposed 1D-CRNN with weight/data bit width quantized to 8/8 bits. Therefore unified training and optimization framework can be performed for various application scenarios and input loads. The proposed 1D-CRNN based keywords recognition system can achieve a higher recognition accuracy with reduced computation operations. Based on system-architecture co-design, an energy-efficient DNN accelerator which can be dynamically reconfigured to process the 1D-CRNN with different configurations is proposed. The processing circuits of the accelerator are optimized to further improve the energy efficiency using a fine-grained precision reconfigurable approximate multiplier. Compared to the state-of-the-art architectures, this work can support 1∼5 real-time keywords recognition with lower power consumption, while maintaining higher system capability and adaptability.**

***Index Terms*—Noise-robust keywords recognition, feature extraction, precision reconfigurable, approximate multiplier**

I. Introduction

The automatic speech recognition enables non-contact interaction between users and devices, which is one of the important signs of machine intelligence. Keywords recognition, also known as keywords spotting, is the most widely used category of speech recognition in the IoT. Keywords recognition is used to identify the specific operation commands (instruction words) or the wake-up word, such as the wake-up words "Hello-xiaona" of Windows-10, and the operation commands "Xiaoai-tongxue", "Previous-song" and "Next-song" of the Xiaomi smart speaker. Since the keywords recognition system is usually required to be always-on and always listening, the ultra-low power and real-time processing with high recognition accuracy are the critical requirements. In the past decades, deep neural networks (DNNs) have demonstrated a more prominent advantage in speech recognition than traditional models (i.e.,the Hidden Markov models and the Gaussian mixture models [1], [2]). The DNN based keywords recognition system is usually composed of a feature extraction module to convert the speech signal into a sequence of acoustic feature vectors, and a keywords classification module using various DNNs.

For decades, many different feature extraction algorithms have been proposed for speech recognition, including the mel-scale frequency cepstral coefficients (MFCC) [3], the linear predictive coefficients (LPC) [4], the perceptual linear production (PLP) [5] and the relative spectral analysis perceptual linear prediction (RASTA-PLP) [6]. In the hardware implementations of DNN based keywords recognition system [7]–[12], although the DNN topologies and settings used are ever-changing and different, they almost all use MFCC as the feature extraction approach based on recognition accuracy and hardware overhead considerations. These works have disadvantages in two aspects: on one hand, the feature extraction and the DNN based keywords classification are two independent modules, therefore the training and optimization of DNN cannot improve the adaptability of feature extraction to target application scenarios. On the other hand, although the power consumption of MFCC accounts for up to 50% of the total power [13], the MFCC algorithm is complex and difficult to optimize, the current works are mainly focused on the optimization of DNN topology and hardware accelerator. The above two problems severely limit the accuracy improvement and power optimization of the keywords recognition processor.

In this paper, we proposed a one-dimensional convolutional recurrent neural network (1D-CRNN) based keywords recognition system with 8/8 bits quantized weight/data bit width. This system can process both the feature extraction and keywords classification, therefore the entire keywords recognition system can be customized and optimized with the DNN training framework to adapt to different application scenarios (i.e., background noise types, SNRs). To accelerate the 1D-CRNN and make it energy efficient, a reconfigurable DNN accelerator is proposed to process the 1D-CRNN with different configurations for various applications. The processing circuits of the accelerator are optimized to further improve the energy efficiency using a fine-grained precision reconfigurable approximate multiplier. This processor can support 1∼5 real-time keywords recognition with lower power consumption, while maintaining higher system capability and adaptability.

II. 1D-CRNN for Noise-robust Keywords Recognition

As shown in Fig. 1, the typical DNN based keywords recognition processor is composed of a speech feature extraction module and a DNN based keywords classification module.

978-3-9819263-5-4/DATE21/©2021 EDAA

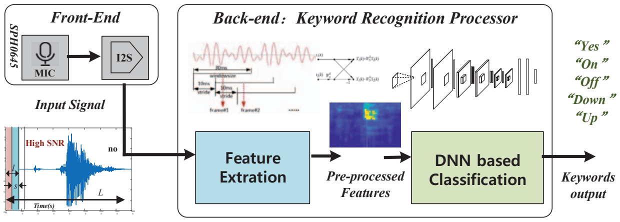


Fig. 1. Typical DNN based keywords recognition processor

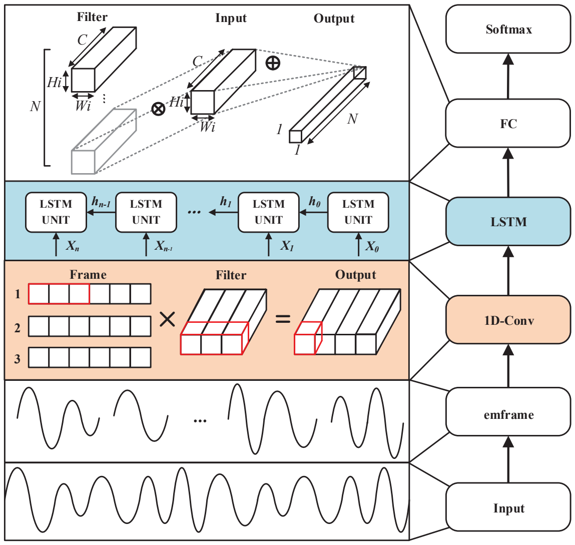


Fig. 2. 1D-CRNN network forward topology

However, once the algorithm of feature extraction module is determined, its hardware implementation is also fixed, such as the MFCC module. Only the DNN topology and parameter settings of the keywords recognition module can be trained and optimized according to the target applications. In the past decades, many DNNs have been proposed for keywords recognition, including the convolutional neural networks (C-NN) [14], the long and short term memory (LSTM) based recurrent neural network (RNN) [15], the gated recurrent unit (GRU) based RNN [16], and the convolutional recurrent neural network (CRNN) [17]. In this work, an optimized 1D-CRNN is proposed which can process both the feature extraction and the keywords classification.

The overall structure of the proposed 1D-CRNN approach for keywords recognition is shown in Fig. 2. Instead of traditional speech feature extraction methods, such as the MFCC, LPC and so on, the input speech signal from the microphone (SPH0645, a digital silicon microphone used in this work) is directly fed into the one-dimensional convolutional layer (1D-Conv) of the 1D-CRNN. The 1D-CRNN first performs the dimensionality reduction on the input speech in the frequency domain, then sends the results to the LSTM layer to extract the features in the time domain, and finally obtains the classification results through the fully-connected (FC) layer and the soft-max layer. Unlike the traditional CNNs, where two-dimensional convolutional layers are used to reduce the dimensionality of input data, in this work we use one-dimensional convolutional layer instead of traditional MFCC to extract the features of input speech, because the spectral characteristics of the input speech are also one-dimensional.

Since the proposed 1D-CRNN implemented on the software platform is simulated in floating-point,firstly we need quantize it to reduce the data/weight bit width for reducing the hardware resources required while implementing the 1D-CRNN. In work [18], a bit-by-bit layer-by-layer quantization method has been proposed for speech recognition which can quantize the bit width of weights to avoid recognition accuracy loss. Based on this method, we quantize the bit width of both data and weights in the proposed 1D-CRNN. We use the Google speech commands dataset (GSCD) [19] as the training and validation set. There are 105K 1-second long audio clips of 35 keywords in the dataset. The proposed 1D-CRNN based keywords recognition models are trained to classify the input speech frames into one of the 5 keywords, "Down", "Up", "Yes", "On", "Off", along with "unknown". The background noises are randomly selected from Noise-92 database [20] including Babble/Pink/White, and then added to the training and testing speeches with SNR from -5dB to Clean. In this work, we quantize the bit width of weight/data to 8/8 bits, which can maintain the high recognition accuracy.

The structures and parameter settings of the 1D-CRNN are required to be further evaluated and optimized to reduce the hardware resources and power consumption required as much as possible. Based on the evaluation of the trade-off between recognition accuracy and hardware overhead requirements, we can find the optimal 1D-CRNN settings. Taking the high accuracy 5 keywords recognition under high background noise environment (from -5dB to clean) as an example, we chose 5 1D-Conv layers, 1 LSTM layer and 1 FC layer to build the 1D-CRNN. The settings and hyperparameters of each layer of the 1D-CRNN are shown in Table I. The accuracy comparisons between the 1D-CRNN based keywords recognition system and the traditional keywords recognition system (based on MFCC and CRNN) are shown in Table II. The CRNN consists of 1 Conv layer(10×4×28, with the stride of 2×2), 2 LSTM layers (30 cells) and 1 FC layer (64 cells). The comparison results show that even for the high background noise of white noise with SNR of -5dB, the recognition accuracy of the 1D-CRNN based keywords recognition system can achieve up to 86%, which is far better than the traditional MFCC and CRNN based approach. Besides, we also use a random mixed data set with various background noise types with SNR from -5dB to 20dB to evaluate the robustness of the proposed 1D-CRNN based keywords recognition system in a wide range of varying noise scenarios. In such a test environment, our system can obtain 87.3% recognition accuracy, which is much higher than traditional solutions.

TABLE I
1D-CRNN OPOLOGIES AND SETTINGS FOR KEYWORDS RECOGNITION

Layer	Kernel	O-Maps	Stride	Units
1D-Conv1	1×5	8	2	NA
MAX POOLING	5	8	2	NA
1D-Conv2	1×10	16	2	NA
1D-Conv3	1×5	32	1	NA
1D-Conv4	1×10	32	2	NA
1D-Conv5	1×5	48	2	NA
AVE POOLING	NA	NA	NA	NA
LSTM	NA	NA	NA	50
FC	NA	NA	NA	5

TABLE II
COMPARISONS OF THE 1D-CRNN BASED KEYWORDS RECOGNITION AND THE TRADITIONAL ARCHITECTURE

Noise Type	SNR(dB)	Accuracy (MFCC+CRNN)	Accuracy (1D-CRNN)
Clean		94.3 %	93.8 %
Pink	10	91.4%	93.6 %
	0	86.1%	91.1%
	-5	78.5%	89.3 %
Babble	10	90.2%	92.3%
	0	81.2%	91.3%
	-5	67.3%	87.6%
White	10	91.3%	90.3%
	0	87.4%	88.3%
	-5	76.8%	86.0%
Random	-5 to 20 mixed	80.9%	87.3%

III. NOISE-ROBUST 1D-CRNN BASED KEYWORDS RECOGNITION PROCESSOR

A. Reconfigurable 1D-CRNN Accelerator for Power-constrained Keywords Recognition

The implementation of a typical hardware implementation architecture for hybrid DNN where different layer types are adopted is shown in Fig. 3(a), which is a distributed computing architecture. In this architecture, two SRAM blocks are used as ping-pong buffers to cache outputs of the convolution computing units and the LSTM computing units. The total SRAM blocks required for this architecture is 52KBytes, as shown in Fig. 3(a). This architecture uses separate modules to process Conv layers and LSTM layers respectively. For each input speech frame, the Conv layers require a total of 190.2K calculation cycles per second, and the LSTM layers require 171.5K calculation cycles per second. There is a bandwidth mismatch between the two modules. In order to improve the hardware resource utilization and reduce the leakage power, we present a reconfigurable 1D-CRNN accelerator architecture based on the typical computing architecture. This reconfigurable architecture uses a configurable PE_Array to process Conv layers and LSTM layers with different configurations. As shown in Fig. 3(b), it consists of 4 weight SRAM blocks with the size of 40bits×1942, a processing elements array (PE-Array) which contains 20 processing elements (PEs), one Nonlinear module, two buffers with the size of 32bits×800 and a Conv Output Buffer with the size of 8bits×1680. The bit width of weight SRAM blocks is set to 40, so that 20 weights can be loaded in parallel the same. As a result, when the PE_array is reconfigured to compute in the LSTM mode, the hardware resource utilization of PE-Array can be maximized.

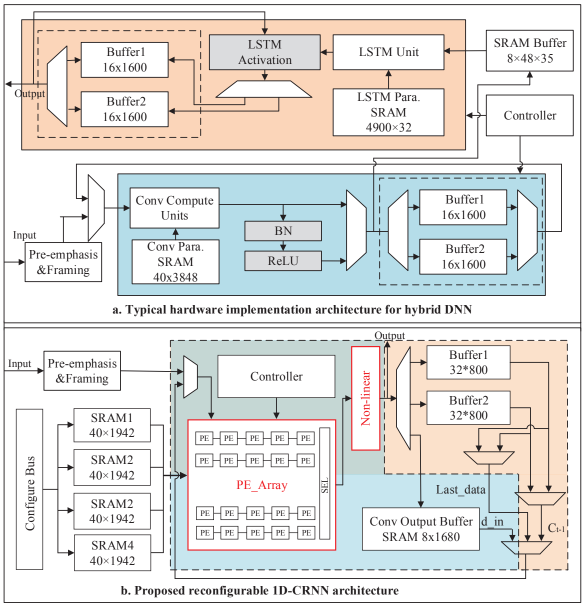


Fig. 3. Overall architecture of the reconfigurable 1D-CRNN accelerator

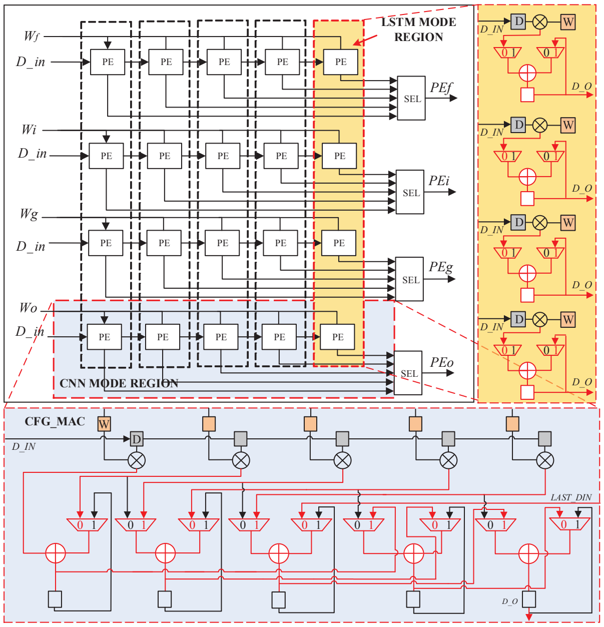


Fig. 4. PE-Array architecture

As shown in Fig. 4, the PE-Array consists of four Configurable Multiply Accumulate (CFG_MAC) units. Take the CNN MODE REGION as an example, the upper part presents 10 regs loaded with *Wo* and *D_in* and 5 multipliers. This part calculates the multiplication of 5 sets of input data. The lower part processes the scheduling of 5 sets of results. With the configuration of the 9 MUXs and the 5 registers, we can achieve an addition tree or an accumulation unit. With the 0/1 selection of the MUXs, the CFG_MAC unit can be reconfigured to process the Conv layers, FC layers or LSTM layers. Taking the CNN MODE REGION as an example, when the selector is set to (0,1,0,1,0,1,0,1,0) from the left to the right, the CFG_MAC unit is configured to multiply and add 5 groups of the input data. If the convolution kernel is 1×10, the output of the units can transmitted to the LAST_DIN port of the next CFG_MAC unit. Taking the LSTM MODE REGION as an example, when the selector is set to (1,0,1,0,1,0,1,0,1), the CFG_MAC unit can complete five sets of multiplications respectively, store them in the bottom regs to realize the accumulation operation, then the LSTM layers or FC layers can be processed.

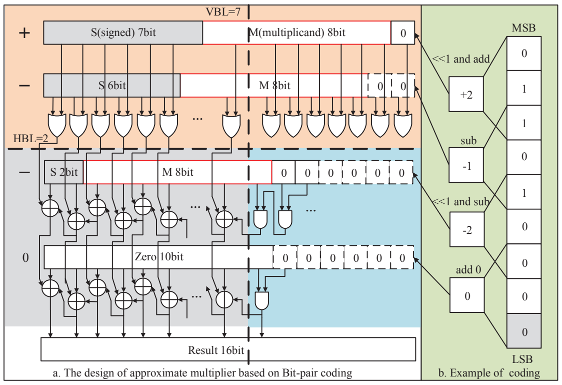
In this work, 4 column-oriented multiplication-accumulation units are used to implement the parallel operation of the four gates F (Forgotten Gate), I (Input Gate), G (Update Gate), and O (Output Gate) of the LSTM unit. The multiplication-accumulation results are: *PEf*, *PEi*, *PEg*, *PEo*, which will be then directly sent to the Non_linear module for activation, avoiding buffering intermediate results. The Non-linear operation module is also reconfigurable, which mainly includes a comparator and a multiplication-accumulation array.

When input speech data transmitted to the accelerator through pre-emphasis and framing module, the PE_Array is set to the CNN mode. The data stream passes through PE_Array and Non-linear modules, and the result is stored in Conv Output Buffer. After all the convolutional layers are processed, the results are stored in the Conv Output Buffer. Then the PE_Array is set to the LSTM mode. The input of PE_Array is provided by the combination of Conv Output Buffer and Buffer2 (or Buffer1). PE_Array loads partial weights of the LSTM layer. When one time-step is processed, the result is stored in Buffer1 (or Buffer2). After the last time-step of the LSTM layer, PE_Array is set to the FC mode, the input data is loaded from Buffer1 (or Buffer2), and the output result is the recognition result. Compared to the typical hybrid DNN architecture (as shown in Fig. 3(a)), the proposed architecture (as shown in Fig. 3(b)) can improve the hardware resources utilization by up to 36%.

B. Fine-grained precision reconfigurable approximate multiplier for 1D-CRNN Computing

In the proposed 1D-CRNN, the operation numbers of additions and multiplications are almost the equal, however the power consumption of multiplications can account for over 90% of all. Thus, a convincing idea to reduce power consumption for processing DNNs is to improve the energy efficiency of multiplication operations. In this work, we present a fine-grained precision reconfigurable approximate multiplier to further reduce the computing power consumption.

The principle of multiplication is essentially shift accumulation, and the number of times the multiplier needs to accumulate is the bit width of the multiplier. In order to reduce power consumption, the proposed approximate multiplier with fine-grained precision reconfigurable architecture can achieve different calculation precisions with reduced circuit power consumption by using two parameters: the Horizontal Breaking Line (HBL) and the Vertical Breaking Line (VBL). The HBL = *n* indicates that the horizontal carry of the *n* rows partial product above HBL is discarded, which indicates the full adder can be replaced by a 2-input OR gate. Similarly, the VBL = *m* means that the vertical carry of the *m* columns partial product on the right-side VBL is discarded, which means the full adder can be replaced by a 2-input AND gate. When the value of HBL and VBL increases, the number of transistors required for the multiplication decreases, and the accuracy loss of the proposed 1D-CRNN may increase. Specifically, if HBL=0 and VBL=0, it will be configured as a standard full precision multiplier.



a. The design of approximate multiplier based on Bit-pair coding

b. Example of coding

Fig. 5. Fine-grained precision reconfigurable approximate multiplier

In this work, we first use booth encoding method for the multiplier to reduce the number of additions and subtractions required for the multiplication. The encoding format is shown in Table III. Before encoding, it needs to add a zero to the lowest bit of the multiplier. The m_{i+1}, m_i, and m_{i-1} respectively represent the adjacent three digits of the multiplier. Each time the coding starts from the lowest bit to the highest bit, each time *i* increases by 2. For every interval of 2 bits, 3 bits of the multiplier are taken to encode, and the operation that needs to be done is achieved this time.

Fig. 5(b) gives an example, which is the encoding of 0110_1000, representing 104. The bit-pair encoding result is {+2,-1,-2,0}, and the decimal system is represented as shown in the Equation (1).

$$d = 2 \times 4^3 - 4^2 - 2 \times 4^1 = 104 \tag{1}$$

For the numbers of which the multiplier data bit width is not even, it is necessary to fill a sign bit in the highest bit, and then perform bit pair encoding. The weight bit width in this paper is 8, through booth encoding the multiplicand only needs to be accumulated 4 times each time. One example of the approximate multiplication implementation is also shown in Fig. 5, where the data bit width of the multiplicand is 8bits. The VBL designed in the schematic diagram is 7, the HBL is 2, and the shaded part is filled with the sign bit.

TABLE III
ENCODING FORMAT FOR PROPOSED APPROXIMATE MULTIPLIER

$\{m_{i+1}, m_i, m_{i-1}\}$	Code	Operation
000	0	Add 0
001	+1	Add multiplicand
010	+1	Add multiplicand
011	+2	<<1 bit and add multiplicand
100	-2	<<1 bit and sub multiplicand
101	-1	Sub multiplicand
110	-1	Sub multiplicand
111	0	Add 0

TABLE IV
RECOGNITION ACCURACY WITH DIFFERENT APPROXIMATE MULTIPLIER SETTINGS (5 KEYWORDS RECOGNITION)

Layer	OPS/%	HBL,VBL			
		CASE1	CASE2	CASE3	CASE4
1D-Conv1	0.88	0,0	0,0	0,0	0,0
1D-Conv2	7.15	0,0	0,0	0,0	0,0
1D-Conv3	14.30	0,0	2.8	3,6	3,8
1D-Conv4	28.60	2,5	2,6	2,5	3,6
1D-Conv5	10.74	0,0	2,8	3,6	3,8
LSTM	38.32	2,5	2,6	2,5	3,6
FC	0.01	0,0	0,0	0,0	0,0
Loss Accuracy		0.5%	1.1%	3.2%	5.6%

For the upper part of the HBL line, since the horizontal carry is ignored, the OR gate can be used. In the lower right corner where HBL and VBL intersect, since the longitudinal carry is ignored, only one AND gate is needed. The lower left corner where HBL and VBL intersect adopts a full adder. This can greatly reduce the resource consumption and power consumption of the multiplier.

Based on the algorithm designed in this work, the recognition accuracy of the network is determined by the accuracy of approximate multiplier. It can be seen that the accuracy of the multiplier has a total of two parameters, HBL and VBL, to adjust the accuracy of the approximate multiplier. When HBL and VBL are smaller, the accuracy is higher. Applying the above-mentioned approximate multiplying adder optimization scheme to the multiplication-accumulation circuit part of the neural network, the approximate scheme and the final accuracy results can be obtained through software simulation as shown in Table IV. OPS% is the ratio of the number of multiplications in the corresponding layer to the total. With the increase of HBL and VBL from CASE1 to CASE4, the loss of precision also increases, and the corresponding power consumption decreases. In this work, we adopt the CASE1 configuration in which the approximate multipliers are only adopted in the layers with the largest amount of calculations. Approximate calculations are performed on the network after quantization, so the degree of approximation that can be taken is limited. The weight bit width after quantization in this paper is set to 8bits, and the data bit width is set to 8bits too. Compared with the full precision standard multiplier used in the process library, the approximate multiplier proposed in this paper with CASE1 configuration can save about 35% power consumption of the PE-Array.

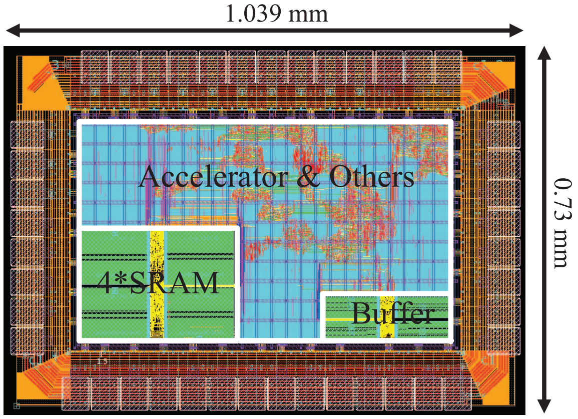


Fig. 6. Layout of the prototype keywords recognition processor

TABLE V
COMPARISONS WITH THE STATE-OF-THE-ART LOW POWER KEYWORDS RECOGNITION PROCESSORS

	TCAS-I'20 [18]	VLSI'18 [10]	ESSCIRC'18 [21]	This work
Technology	22nm	65nm	65nm	22nm
Architecture	MFCC+BWN	MFCC+BCNN	MFCC+LSTM	1D-CRNN
Bit width (Data)	MFCC: 12/16bits BWN: 16bits	MFCC: 16bits BCNN: 1bit	MFCC: 16bits LSTM: 16bits	LSTM: 9bits
Bit width (Weight)	1bit	1bit	16bits	8bits
Frequency	250KHz	2.5MHz	250KHz	250KHz
Latency	16ms	0.5∼25ms	16ms	16ms
Voltage	0.6V	0.57V	0.57V	0.6V
Layout Area	0.602mm^2	9.611mm^2	1.035mm^2	0.758mm^2
Memory	11KB	52KB	34KB	46KB
Numbers of Keywords	10	1	4	1∼5
Power	10.8∼15.1μW	141μW	5μW*	1.8μW@1-keyword 2.7μW@5-keywords
Dataset	GSCD	TIDIGIT	NA	GSCD
Recognition Accuracy	87.9%@Clean 84.4%@10dB 80.8%@5dB	95%@Clean 88%@10dB	91.2%@Clean	5-Keywords: 93.2%@Clean 91.5%@10dB 87.1%@-5dB 1-Keyword: 98.6%@Clean 97.4%@-5dB

(*5μW in work [21] does not include the power consumption of the MFCC module for feature extraction.)

IV. IMPLEMENTATION RESULTS

To evaluate the power consumption and recognition accuracy of the proposed keywords recognition processor, the prototype processor shown in Fig. 3 (b) is implemented and evaluated on an industrial 22nm ultra-low-leakage (ULL) process technology. The prototype system is functional with the logic supply voltage of 0.6V, and the clock frequency is 250KHz. The power consumption is evaluated with Synopses PTPX at 25°C TT corner. The layout of the proposed 1D-CRNN based keywords recognition processor using 22nm ULL technology is shown in Fig. 6. The area of the prototype system is 0.758 mm^2 (1.039mm × 0.730mm) with the I/O PADs (0.331 mm^2 without the I/O PADs).

Comparisons with the state-of-the-art designs for keywords recognition processors are shown in Table V. In these works, although they use different DNN structures to implement keywords classification, they all use a MFCC module for the feature extraction. Using more bit width for MFCC module can improve the recognition. However, accuracy of the keywords recognition system to a certain extent for the wide-ranged background noise applications, the recognition accuracy will quickly decrease as the SNR decreases, as discussed above. As shown in Table V, when the proposed 1D-CRNN based keywords recognition processor is configured for 1-keyword recognitor (configured to 3 1D-Conv layers, 1 LSTM layer and 1 FC layer), this work can achieve the accuracy of 98.6%@Clean and 97.4%@-5dB, with the power consumption of 1.8 μW; when the 1D-CRNN based keywords recognition processor is configured for 5-keywords recognitor (configured to 5 1D-Conv layers, 1 LSTM layer and 1 FC layer), this work can achieve the accuracy of 93.2%@Clean, 91.5%@10dB and 87.1%@-5dB, with the power consumption of 2.7 μW. Compared to state-of-the-art designs, the proposed 1D-CRNN based keywords recognition processor can process both the feature extraction and keywords classification without MFCC, and can achieve noise-robust high accuracy with low-power consumption.

V. Conclusion

In this paper, a low power reconfigurable keywords recognition processor is implemented with high accuracy under 22nm CMOS technology. The proposed 1D-CRNN with 8/8 weight/data bit width quantization can efficiently process both the feature extraction and keywords classification without MFCC under different noise and SNRs. Besides, an energy-efficient DNN accelerator using fine-grained precision reconfigurable multipliers is proposed to process the 1D-CRNN with different configurations for different workloads and various applications. The proposed keywords recognition processor can support noise-robust high accuracy 1 $\sim$ 5 keywords recognition with the power consumption of 1.8μW $\sim$ 2.7μW.

Acknowledgment

This work is funded with the National Science and Technology Major Project under Grant 2018ZX01031101-005 and National Natural Science Foundation of China under Grant 61904028.

References

[1] M. L. Seltzer, et al., “An investigation of deep neural networks for noise robust speech recognition,” in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, Vancouver, BC, 2013, pp. 7398–7402, 10.1109/ICASSP.2013.6639100

[2] G. E. Dahl, et al., “Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, Jan. 2012, 10.1109/TASL.2011.2134090

[3] 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,” *Journal of Computer and Communications*, 2015, vol. 3, pp. 1–9, 10.4236/jcc.2015.36001

[4] P. B. Patil, “Multilayered network for LPC based speech recognition,” *IEEE Transactions on Consumer Electronics*, 1998, vol. 44, no. 2, pp. 435–438, 10.1109/30.681960

[5] H. Hermansky, “Perceptual linear predictive (PLP) analysis of speech,” *The Journal of the Acoustical Society of America*, 1990, vol. 87, no. 4, pp. 1738–1752, 10.1121/1.399423

[6] H. Hermansky, et al., “The Challenge of Inverse-E: The RASTA-PLP Method,” *Conference Record of the Twenty-Fifth Asilomar Conference on IEEE*, 1991, pp. 800–804, 10.1109/ACSSC.1991.186557

[7] S. Bang, et al., “A 288 μW programmable deep-learning processor with 270kb on-chip weight storage using non-uniform memory hierarchy for mobile intelligence,” *IEEE International Solid-state Circuits Conference*, San Francisco, CA, USA, 2017, pp. 250–251, 10.1109/ISSCC.2017.7870355

[8] M. Price, et al., “A scalable speech recognizer with deep-neural-network acoustic models and voice-activated power gating,” *2017 IEEE International Solid-State Circuits Conference*, San Francisco, CA, USA, 2017, pp. 244–245, 10.1109/ISSCC.2017.7870352

[9] M. Shah, et al, “A fixed-point neural network architecture for speech applications on resource constrained hardware,” *J. Signal Process. Syst.*, 2018, vol. 90, no. 5, pp. 727–741, 10.1007/s11265-016-1202-x

[10] S. Yin, et al., ”A 141 μW, 2.46 pJ/neuron binarized convolutional neural network based self-learning speech recognition processor in 28nm CMOS,” *2018 IEEE Symposium on VLSI Circuits*, Honolulu, HI, USA, 2018, pp. 139–140, 10.1109/VLSIC.2018.8502309

[11] S. Zheng, et al., “An Ultra-Low Power Binarized Convolutional Neural Network-Based Speech Recognition Processor With On-Chip Self-Learning,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2019, vol. 66, no. 12, pp. 4648-4661, 10.1109/TCSI.2019.2942092

[12] P. Li and H. Tang, “Design of a Low-Power Coprocessor for Mid-Size Vocabulary Speech Recognition Systems,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2011, vol. 58, no. 5, pp. 961-970, 10.1109/TCSI.2010.2090569

[13] J. S. P. Giraldo, et al., “Vocell: A 65-nm Speech-Triggered Wake-Up SoC for 10-μW Keyword Spotting and Speaker Verification,” *IEEE Journal of Solid-State Circuits*, vol. 55, no. 4, pp. 868–878, 2020, doi: 10.1109/JSSC.2020.2968800

[14] L. Toth, “Combining Time-and Frequency-Domain Convolution in Convolutional Neural Network-Based Phone Recognition,” *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, ltaly, 2014, pp. 190–194, 10.1109/ICASSP.2014.6853584

[15] M. Sun, et al., “Max-pooling loss training of long short-term memory networks for small-footprint keyword spotting,” *2016 IEEE Spoken Language Technology Workshop (SLT)*, San Diego, CA, USA, 2016, pp. 474–480, 10.1109/SLT.2016.7846306

[16] Y. Zhang, et al., “Hello Edge: Keyword Spotting on Microcontrollers,” 2018, (arXiv:1711.07128v3)

[17] S.O. Arik, et al., “Convolutional recurrent neural networks for small-footprint keyword spotting,” 2017, (arXiv:1703.05390)

[18] B. Liu, et al., “A 22nm, 10.8uW/15.1uW Dual Computing Modes High Power-Performance-Area Efficiency Domained Background Noise Aware Keyword-Spotting Processor,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2020, vol. 67, no. 12, pp. 4733–4746, 10.1109/TCSI.2020.2997913

[19] P. Warden. “Speech commands: A dataset for limited-vocabulary speech recognition,” 2018, (arXiv:1804.03209)

[20] A. Varga, et al., “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 Communication*, 1993, vol. 12, pp. 247–251, 10.1016/0167-6393(93)90095-3

[21] J. S. P. Giraldo and M. Verhelst, “Laika: A 5μW Programmable LSTM Accelerator for Always-on Keyword Spotting in 65nm CMOS,” *IEEE 44th European Solid State Circuits Conference (ESSCIRC)*, 2018, pp. 166–169, 10.1109/ESSCIRC.2018.8494342