A 250.3µW Versatile Sound Feature Extractor Using 1024-point FFT 64-ch LogMel Filter in 40nm CMOS

Akiho Kawada
School of Engineering
The University of Tokyo
Tokyo, Japan
akihokawada@g.ecc.u-tokyo.ac.jp

Kenji Kobayashi
Graduate School of Engineering
The University of Tokyo
Tokyo, Japan
kobayashi-kenji@g.ecc.u-tokyo.ac.jp

Jaewon Shin
Graduate School of Engineering
The University of Tokyo
Tokyo, Japan
jwshin@g.ecc.u-tokyo.ac.jp

Rei Sumikawa
Graduate School of Engineering
The University of Tokyo
Tokyo, Japan
sumi13rei@g.ecc.u-tokyo.ac.jp

Mototsugu Hamada
Graduate School of Engineering
The University of Tokyo
Tokyo, Japan
hamada@dlab.t.u-tokyo.ac.jp

Atsutake Kosuge
Graduate School of Engineering
The University of Tokyo
Tokyo, Japan
kosuge@dlab.t.u-tokyo.ac.jp

***Abstract*—A 250.3µW always-on sound feature extractor that facilitates general-purpose sound recognition AI processing encompassing 35-word voice command recognition, environmental sound recognition, and musical instrument recognition is developed. Conventionally, approximated mel-frequency cepstrum coefficients (MFCC) feature extractors composed of a limited number of FFT samples (256 points), and filter channels (10 channels) are utilized for energy reduction; however, their applicability is restricted to wake-up word recognition resulting in high NRE costs. To overcome these challenges, we developed a LogMel filter feature extractor employing a 1024-point FFT and 64-channel Mel filter bank, which enables versatile applications across a diverse range of sound recognition tasks including 35-word voice command recognition. To minimize circuit area and power consumption, three techniques are employed: (a) radix-2^2 single-path delay feedback (R2^2SDF) which uses serial FFT processing for circuit area reduction, (b) zero-skipping Mel filter bank for a 1/25x circuit area reduction by storing and accumulating only non-zero elements, and (c) Log LUT, an LUT approximation to reduce the number of cycles by a factor of 20 compared with the CORDIC implementation. Designed and implemented in a 40nm CMOS process, the proposed extractor demonstrates a power efficiency of 14.9nJ/frame/word for a 35-word voice command recognition task, showcasing a 1.1× improvement in power efficiency and a 17.5× increase in the number of recognizable voice commands compared to state-of-the-art KWS-specific simplified MFCC audio extraction circuits.**

Keywords—Sound processing, MFCC, LogMel, FFT, IoT

I. Introduction

Advances in speech recognition AI have significantly propelled the social implementation of speech information processing technologies. This has led to a growing interest in always-on voice applications. Their specific use cases include single wake-up word recognition known as keyword spotting (KWS) [1-4], environmental sound recognition, and sound recognition [5]. In recent years, there has been a surge in research on voice interface technology that can recognize a diverse range of commands and control wearable devices and drones. For these applications, it is essential to go beyond simple single-word keyword recognition and continuously recognize a larger set of commands, such as 30 words or more [3, 4]. Environmental sound recognition technology is utilized for detecting machine abnormalities, among other applications. Such diverse sound recognition AI models are expected to be processed on battery-powered wearable IoT devices. AI processors for continuous audio analysis must operate at or below 1mW [8] to ensure minimal impact on battery life in wearable devices.

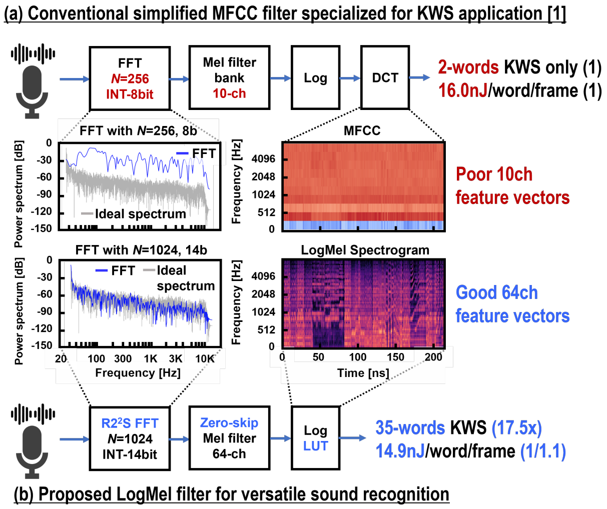


Fig. 1 Sound feature extractor comparison
(a) 10ch MFCC with 256-point FFT and
(b) our 64ch LogMel with 1024-point FFT.

Existing sound recognition AI models employ a large number of FFTs and consume a lot of power. In conventional audio recognition software implementations, 1024-point FFT and 64-channel Mel filter bank have been widely used [5] to extract high-dimensional feature vectors that adequately represent speech signals. Implementing such a high-resolution FFT requires a significant number of computation blocks. For 1024-point FFT, the FFT core alone requires a chip area of 8.3 mm² and consumes 3.7 mW at 30 MHz in 65 nm CMOS technology [6].

In pursuit of audio analysis below, AI processors have been developed that employ highly simplified audio feature extractors and binary weight DNNs (Fig. 1 (a)). For speech recognition tasks, sound feature extractors are widely used that extract a 2D feature map from sound information and feed its output to a subsequent DNN. However, the implementation of these sound feature extractors consumes large power mainly due to the high computational cost of FFT compared to DNNs [1]. To reduce power and area, task-specific sound feature extractors have been proposed in [1-3]. These approaches achieve good power efficiency by drastically reducing the number of FFT points from 1024 to 256 or fewer and minimizing the number of band-pass filters from 64 to 10, tailoring the design to small and specific tasks. However, their applicability is limited due to the insufficient feature

Table I Performance comparison on KWS task

	Ref. [1]	Ref. [3]	Ours:
# of keywords	2 (1)	30	35 (17.5x)
FFT Samples, *N*	256 (1)	512	1024 (4x)
Mel-filter channels	10 (1)	40	64 (6.4x)
Energy efficiency at nominal supply voltage [nJ/frame/word]	16.0*1 (1)	8800.0*2	14.9 (1/1.1)

extraction for generalizability across different applications and the inability to adapt to other tasks, leading to high non-recursive engineering (NRE) costs.

This research introduces a general-purpose sound feature extractor that employs a LogMel filter to achieve low NRE cost and low power consumption below 1mW simultaneously. For example, our chip can accommodate additions or changes to voice commands without changing the hardware, resulting in low NRE costs while consuming less than 1mW. Featuring a 1024-point FFT and 64-channel Mel filter bank, it facilitates versatile applications including 35-word speech command recognition, environmental sound recognition, and musical instrument recognition [5] (Fig. 1 (b)). We skip the discrete cosine transform that is performed in MFCCs and output the result directly. This allows us to obtain feature maps with more information content, as the frequency-domain features are not compressed. It is known that using the same CNN, higher accuracy can be achieved with LogMel features compared to MFCCs [7]. Although the noise reduction ability of LogMel features is lower than that of MFCCs, this is not a problem because CNNs have high noise immunity due to max pooling layers.

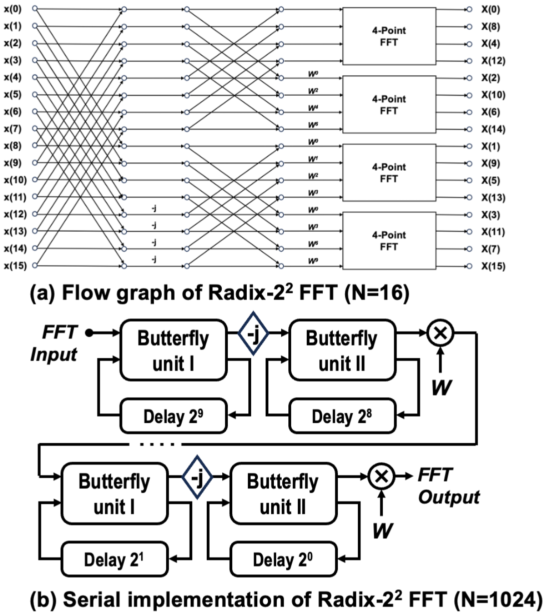
Designed in 40nm CMOS, our test chip exhibits a power consumption of 250.3μW, demonstrating a 1.1 × improvement in energy efficiency per keyword compared to the sparse sound feature MFCC filter optimized for keyword recognition (Table I). When combined with a sound-specific AI processor [4] that processes 16-layer audio DNNs in 14 bits, it is possible to process a wide-range of sound recognition tasks at 400μW.

II. Proposed LogMel-Based Versatile Sound Feature Extractor

A. $R2^2SDF$ (Radix-2^2 single-path delay feedback)

In this study, the raidx-2^2 FFT computation method and a serial FFT architecture are employed. In the radix-2 and radix-4 algorithms, the FFT is recursively divided into smaller FFT sub-stages consisting of data-reordering butterfly operations and twiddle factor multiplications. This reduces the computational complexity from $O(N^2)$ to $O(NlogN)$. Compared to the radix-2 algorithm, where the radix of the butterfly operation is 2, the radix-4 algorithm can reduce the number of multiplications by 25% and the number of additions by approximately 6%. On the other hand, a drawback of the radix-4 algorithm is that when attempting to implement it in hardware using a serial architecture, the control structure becomes more complex.

The radix-2^2 algorithm was developed in [7] to inherit the simple control structure of radix-2 but achieve identical computational requirements as radix-4. By grouping two stages of the radix-2 algorithm into one set and transforming the equation so that the first twiddle factor multiplication becomes multiplication by $-j$, the radix-2^2 algorithm effectively changes the base of iteration to 4 (Fig. 2 (a)). Owing to the transformation, radix-2^2 has the simple logical circuit structure of radix-2 and the same low computational cost as radix-4. To reduce the circuit area, we employed $R2^2SDF$, a serial implementation of radix-2^2 (Fig. 2 (b)).



(a) Flow graph of Radix-2^2 FFT (N=16)

(b) Serial implementation of Radix-2^2 FFT (N=1024)

Fig. 2 $R2^2SDF$ Circuit Diagram

B. Zero-Skipping Mel-Filter Bank

In a Mel-filter bank, the output of a low-pass filter for each channel is multiplied and accumulated (MAC) with the output of the FFT. By concatenating the MAC results of all the channels, a single feature map can be constructed and output. Each Mel-filter is a low-pass filter based on the characteristics of human hearing, and each filter passes only a specific frequency band (Fig. 3). The characteristics of these filters can be stored in a ROM on the hardware since they are always constant when the number of filters, frequency bands, and the number of FFT points are fixed, and independent of sound recognition tasks.

As the number of filters increases, the ROM capacity and the number of MAC units for parallel processing also increase. In this study, we developed a zero-skipping Mel-filter bank to reduce ROM capacity and computational complexity even when the number of channels increases. Most of the Mel-filter values are zero. Each filter is configured to pass only signals in a specific frequency band. In the specific frequencies corresponding to the FFT, out of 64 filters, at most only 2 filters have non-zero values, while all others are zero. Therefore, we adopted the compressed sparse column (CSC) format, which stores only the non-zero values of the filter along with their filter identifiers and positions within the filter. The CSC format reduces the size of the filter coefficient matrix from 853,632 bits to 33,858 bits, a reduction of about 1/25.2. Since the ROM area is proportional to the size of the filter matrix to be stored, the ROM area is reduced by 25.2.

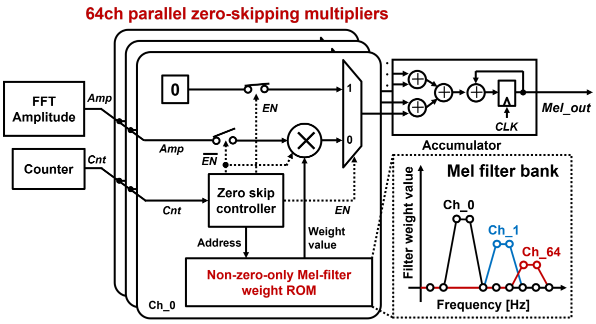


Fig 3. Zero-skipping Mel-filter bank and MAC unit

In addition to the CSC format for reducing ROM size, we also propose a zero-skipping multiplier for efficient computation. When the filter coefficient is a zero element, the data is not moved to the multiplier to reduce power. A constant zero is output. When the filter coefficient is non-zero, an *EN* signal is issued, the multiplier's power is turned on, and the calculation is performed. The output is switched from a fixed zero to the multiplier output.

In LogMel, the filter coefficients for each channel are always constant, and the positions of the non-zero bits are also fixed. Therefore, the controller can remember the positions of the non-zero bits for each channel easily by storing them in ROM, making it possible to reuse the controller for a wide range of applications. The output of the FFT is transmitted serially from low frequency to high frequency, so the timing of the EN signal can be easily detected by using a counter. These processes make it possible to reduce the number of multiplications and power consumption by a factor of 1/25.2, similar to the reduction in filter coefficient matrix size.

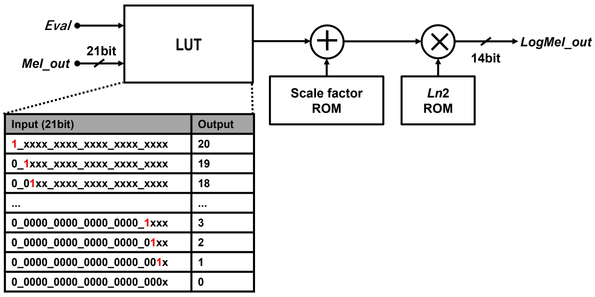
C. Log LUT

A large number of cycles is required for LogMel due to the use of CORDIC circuits for natural logarithm calculations. In LogMel, the output signal of the Mel filter bank passes through a natural logarithm function to produce the final feature map. Natural logarithm calculations are typically performed using CORDIC circuits. Since the accuracy is gradually improved through iterative calculations, a certain number of iterations is required. In previous studies, 20 iterations were needed [9]. At least 1280 cycles would be required for the above case, which is greater than the number of cycles for other computation blocks (serial FFT, zero-skipping Mel filter bank). This causes a performance bottleneck for processing latency.

We propose a natural log circuit using LUT to reduce the number of cycles. First, the natural logarithm calculation is converted to a logarithmic calculation with base 2 (Eq. (1)).

$$Ln(x) = log_2(x) * ln2 \qquad (1)$$

The value of $Ln2$ is precomputed in advance and stored in ROM. The value of $Log_2(x)$ is obtained by looking it up in a table. The LUT uses the position of the highest-order bit set to 1 as its key, and the approximate value of $Log_2(x)$ as its value. Since this requires just referencing the LUT stored in ROM, the number of cycles is drastically reduced from 1280 to 64. For example, for x =00100101 in binary digits (37 in decimal), the value of $Log_2(x)$ is approximated as ≈5. Since the value of $Log_2(37s$ is 5.2, $Log_2(x) \approx 5$ is a good approximation. After calculating $Log_2(x)$, scaling adjustments are performed by adding the scale factor. The factor is constant after setting 1024-point FFT and Mel filter banks and stored in scale factor ROM. The eval signal is issued after finishing computing Mel filter bank MACs to start computing the Log LUT. We compared the results of a 64-bit calculation of $Ln(x)$ for sound data using Python's Numpy library with the results of the 14-bit calculation using the Log LUT method described above. The average error rate was 2.20%. Therefore, good approximation is realized.



Input (21bit)	Output
1_xxxx_xxxx_xxxx_xxxx_xxxx	20
0_1xxx_xxxx_xxxx_xxxx_xxxx	19
0_01xx_xxxx_xxxx_xxxx_xxxx	18
...	...
0_0000_0000_0000_0000_1xxx	3
0_0000_0000_0000_0000_01xx	2
0_0000_0000_0000_0000_001x	1
0_0000_0000_0000_0000_000x	0

Fig 4. Log LUT computation circuit

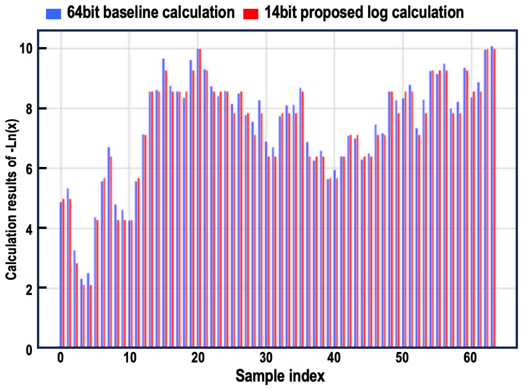


Fig 5. Log LUT approximation simulation results

III. Experimental Results

A. Chip Implementation

The proposed approximated LogMel filter-based versatile sound feature extractor was implemented using 40nm CMOS. The chip implementation area is $2340\mu m \times 174\mu m$ .The number of clock cycles required for the calculation is 2064 cycles. At a clock frequency of 500 kHz, the delay time for LogMel processing is 4.13 ms. The power consumption is 250.3 μW at 500 kHz with a nominal supply voltage of 1.1 V. The processing energy is 521.5 nJ/frame.

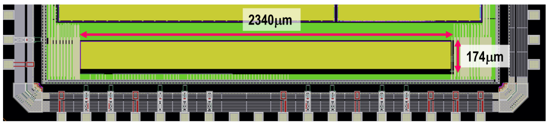
B. Sound Recognition Performance

The inferential performance of the LogMel feature extractor combined with DNN processing was simulated for various sound recognition tasks [5]. The wired-logic processor proposed in Ref. [4] was used as the DNN processor model. The dataset and tasks were environmental sound recognition, musical instrument recognition, and KWS. The training results for each task and a comparison to the baseline software implementation (FP 64bit, Python implementation) are shown in Table-II. The average accuracy drop is only 3.2%. In addition, the recognition accuracy is significantly improved by 16.3 % compared to the feature extractor using a 128-point FFT, which has been used in conventional always-on applications.

Table III Performance comparison with state-of-the arts

	JSSC'21 [1]	JSSC'22 [3]	ISCAS'21 [2]	Ours
Applicable tasks	2-words KWS	12-words KWS	30-words KWS	(1) 35-words-KWS, (2) Language identification (3) Environmental sound
Filter type	MFCC	Analog filter	MFCC	LogMel
Filter output dimensions	10	16	40	64
Points of FFT	*N*=256	NA	N=512	*N*=1024
Process node	28nm CMOS	65nm CMOS	180nm CMOS	40nm CMOS
Feature extractor circuit area	0.054mm^2	1.60mm^2	2.39mm^2	0.41mm^2
Power consumption of feature extractor	2.00 μW*1	9.3 μW	26.4 mW	250.3 μW
Energy efficiency in KWS at normal supply voltage	16.0 nJ/frame/word*1 (1)	12.7 nJ/frame/word	8800.0 nJ/frame/word*2	14.9 nJ/frame/word (1/1.1)

*1: Based on Figure 16, we calculate that the power consumption is 3.03μW in total at the normal supply voltage of 1V, and calculate MFCC power based on Fig. 17 in Ref. [1]. *2: According to the paper, we assume the frame rate is 100fps.



Process	40nm CMOS	Bit width	14bit
Area	2340μm × 174μm	Supply voltage	1.1V
Clock	500kHz	Total power	250.3μW

Fig 6. Chip implementation results.

C. *Performance Comparison*

The results obtained are compared with previously reported keyword spotting recognition ASICs for always-on applications (Table-III). Owing to the 1024-point FFT and the Mel filter bank with a large amount of information, not only KWS with 35 words but also language recognition and environmental sound recognition are possible. The energy efficiency of KWS (14.9 nJ/frame/word) is as good as or better than that of conventional ASIC implementations while realizing low NRE costs. Conventional analog implementation can achieve better energy efficiency, but they are easily affected by PVT variations and require a large implementation area. On the other hand, our processor is pure digital and it is robust against PVT variations and has a smaller chip area.

IV. Conclusions

A 250.3μW always-on sound feature extractor that facilitates general-purpose sound recognition AI processing encompassing 35-word voice command recognition, environmental sound recognition, and musical instrument recognition has been implemented. We developed a LogMel filter feature extractor employing a 1024-point FFT and 64-channel Mel filter bank, enabling versatile applications across a diverse range of sound recognition tasks, including 35-word voice command recognition. Owing to the use of R2^{2}SDF FFT, zero-skipping Mel filter bank and the Log-LUT computing method, our chip satisfies both low-power consumption and low-NRE costs simultaneously. The proposed extractor fabricated in 40 nm CMOS demonstrates a power efficiency of 14.9 nJ/frame/word for a 35-word voice command recognition task, showcasing a 1.1× improvement in power efficiency and a 17.5× increase in the number of recognizable voice commands.

Table II Accuracy Comparison on Sound Recognition

	Feature	SW/HW	FFT sample	Environ mental	Linguistic Recog.	KWS-12words	KWS-35words	Average
Baseline [5]	LogMel	64bit SW	*N*=1024	78.1%	83.5%	86.5%	88.9%	84.3% (1)
Conv.		14bit HW	*N*=256	55.7%	69.6%	75.3 %	58.8%	64.8%
This work		14bit HW	*N*=1024	74.7%	83.3%	85.1%	81.4%	81.1% (-3.2%)

Our general-purpose speech feature extractor realizes low power consumption, high area efficiency, and applicability to a wide variety of applications simultaneously. The NRE cost is low and it is expected to become a fundamental technology for sound interfaces for various kinds of industrial applications.

Acknowledgment

This work was supported by JST, PRESTO JPMJPR21B4.

References

[1] W. Shan *et al*., “A 510-nW Wake-Up Keyword-Spotting Chip Using Serial-FFT-Based MFCC and Binarized Depthwise Separable CNN in 28-nm CMOS,” in *IEEE Journal of Solid-State Circuits*, vol. 56, no. 1, pp. 151-163, Jan. 2021.

[2] L. Wu *et al*., “A High Accuracy Multiple-Command Speech Recognition ASIC Based on Configurable One-Dimension Convolutional Neural Network,” in *IEEE ISCAS*, May 2021.

[3] K. Kim *et al*., “A 23-μW Keyword Spotting IC With Ring-Oscillator-Based Time-Domain Feature Extraction,” in *IEEE Journal of Solid-State Circuits*, vol. 57, no. 11, pp. 3298-3311, Nov. 2021.

[4] R. Sumikawa *et al*., “A183.4-nJ/inference 152.8-μW 35-Voice Commands Recognition Wired-Logic Processor Using Algorithm-Circuit Co-Optimization Technique,” in *IEEE Solid-State Circuit Letters*, vol. 7, pp. 22-25, 2024.

[5] D. Niizumi *et al*., “BYOL for Audio: Self-Supervised Learning for General-Purpose Audio Representation,” in *IEEE International Joint Conference on Neural Networks*, 2021. [Online]. Available: https://arxiv.org/abs/2103.06695

[6] D. Jaeon *et al*., “A Super-Pipelined Energy Efficient Subthreshold 240 MS/s FFT Core in 65 nm CMOS,” in *IEEE Journal of Solid-State Circuits*, vol. 47, no. 1, pp. 23-34, Jan. 2012.

[7] S. He and M. Torkelson, "A New Approach to Pipeline FFT Processor," in IEEE Proceedings of IPPS ’96, 1996, pp. 766 -770.

[8] A. Kosuge *et al*, “A 183.4nJ/inference 152.8uW Single-Chip Fully Synthesizable Wired-Logic DNN Processor for Always-On 35 Voice Commands Recognition Application,” in IEEE Symposium on VLSI Circuits, June 2023.

[9] D. Llamocca and C. Agurto, “A Fixed-point implementation of the natural logarithm based on a expanded hyperbolic CORDIC algorithm”, in XII Workshop IBERCHIP, 2006.