0.08mm^2 128nW MFCC Engine for Ultra-low Power, Always-on Smart Sensing Applications

Yi Sheng Chong*, Wang Ling Goh†, Yew Soon Ong‡, Vishnu P. Nambiar§, Anh Tuan Do¶
*†School of Electrical and Electronic Engineering, Nanyang Technological University (NTU), Singapore
*Energy Research Institute, Interdisciplinary Graduate Programme, NTU, Singapore
‡School of Computer Science and Engineering, NTU, Singapore
§¶Institute of Microelectronics, A*STAR, Singapore
*yisheng002@e.ntu.edu.sg, †ewlgoh, ‡asysong@ntu.edu.sg,
§vishnu_paramasivam, ¶doat@ime.a-star.edu.sg

***Abstract*—Mel frequency cepstral coefficient (MFCC) features are widely used in applications such as keyword spotting, bearing fault detection and heart sound classification. This work proposes a low power MFCC engine that enables its use for battery-powered edge applications. Three hardware algorithm co-optimizations were adopted to achieve energy efficient MFCC hardware implementation. The approximated MFCC features due to the optimizations still allows good accuracy when deployed in several applications such as keyword spotting and bearing fault detection, reporting negligible accuracy drop of $\leq$1.5%. The proposed MFCC hardware consumes only 128nW at 0.3V supply and occupies only 0.08mm^2 in 40nm CMOS technology, which are 5$\times$ and 2.75$\times$ power and area reduction respectively when compared to the prior arts.**

I. Introduction

Mel frequency cepstral coefficient (MFCC) is a well known feature used in many sound, speech or time-series data related applications, such as heart sound classification [1], keyword spotting [2] and machinery health monitoring [3]. The MFCC feature extraction involves two key ideas: Mel frequency scale and cepstrum. The Mel frequency scale is to mimic the human hearing mechanism [4]. While, cepstrum is the information extracted from the spectrum generated by the fast Fourier transform (FFT) [5]. The first use of MFCC features was demonstrated by Davis and Mermelstein in 1980 where they reported a high accuracy of 96.5% in word recognition [6]. Recent work such as [7] and [8] also employed MFCC features with accuracy of over 90% in IoT edge applications.

The MFCC hardware thus has become increasingly important in supporting ultra low-power edge applications [7], [8]. The MFCC feature extraction hardware must consume minimum power to maximize the battery lifetime of such systems from two aspects. First, the MFCC hardware can support always-on features to process any input signal. Second, lower power MFCC hardware contributes to lower system power because the MFCC hardware takes up significant portion of 25% to 67% of the overall system power as reported in work [7] and [9]. Various work for energy efficient MFCC hardware had been reported. For instance, [10] reduced the area and energy consumption of the MFCC in field programmable gate array (FPGA). Analog MFCC-like feature extraction [11] was also explored to bypass the need for an analog-to-digital converter (ADC). Recently, MFCC engines were also customised and integrated to keyword spotting accelerators for tiny edge applications [7], [9].

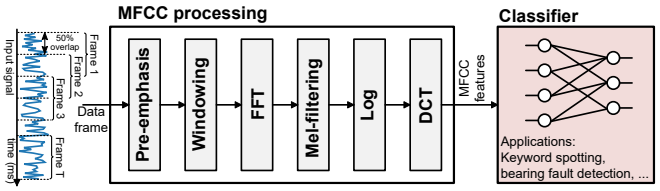


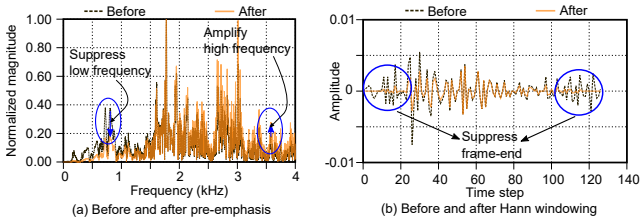
Fig. 1. MFCC processing pipeline. Input signal is divided into frames. Each frame is processed through six steps: pre-emphasis, windowing, FFT, Mel filtering, logarithmic operation and discrete cosine transform (DCT), to produce MFCC features. The MFCC features are used by a classifier for any target applications.

Hence, this work focuses on proposing a fully digital, compact, and low power MFCC hardware to support not only the always-on features, but also for multiple applications that use MFCC features such as keyword spotting, bearing fault detection and heart sound classification. The output MFCC features can be directly fed to a classifier for on-chip classification. The remainder of the paper is as followed. Section II presents our key design features. Section III evaluates our MFCC engine's performance when compared with the states of the arts. Section IV concludes our paper.

II. MFCC and its Hardware Optimization

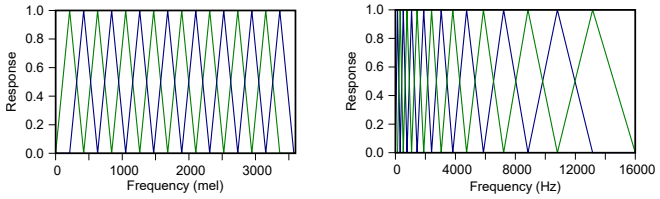
A. MFCC feature extraction

The MFCC engine processes input signal frame by frame, as shown in Figure 1. The signal frame is first pre-emphasized by passing through a high pass filter, to balance amplitude of both the high and low frequency components, as detailed in Figure 2(a). Next, Hanning window is applied to the signal frame to increase the smoothness of the signal between frames, as illustrated in Figure 2(b). Subsequently, the FFT operation is performed to transform time domain input into frequency domain, producing an output vector of N complex elements. Each vector element is the frequency bin, representing the amplitude of the corresponding frequency. The FFT output



(a) Before and after pre-emphasis (b) Before and after Hann windowing

Fig. 2. (a) Effect of pre-emphasis, (b) effect of windowing.



(a) Filters in Mel scale (b) Filters in frequency scale

Fig. 3. Linear gap between filters in Mel scale but varying gap in physical frequency domain.

is squared to obtain the magnitude, followed by Mel filtering. In the Mel filtering step, the frequency is mapped to Mel frequency scale, using Eq. 2, to mimic the human ear hearing perception [12]. The Mel-filtered outputs are then passed through a logarithmic function to mimic the human ear sensitivity. Finally, the discrete cosine transform (DCT) operation is applied to obtain the MFCC features. The DCT involves a dot product between the input signal and a set of cosine coefficients, as shown in Eq. 3. The final MFCC features is used by a following classifier that executes the desired tasks.

$$y(t) = x(t) - \alpha \cdot x(t-1) \tag{1}$$

$$\text{Mel}(f) = 2595 \cdot \log_{10}(1 + \frac{f}{700}) \tag{2}$$

$$X_m = \sqrt{\frac{\beta}{E}} \sum_{n=0}^{E-1} x_n \cos[\frac{\pi}{E}(n + \frac{1}{2})m] \tag{3}$$

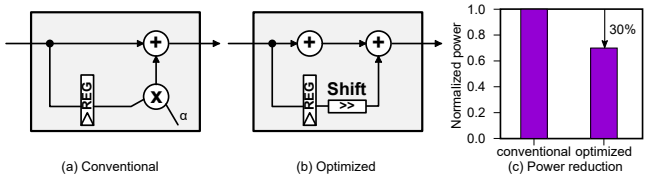
For Eq. 1 to 3, $x(t)$ is the input sample, $y(t)$ is the pre-emphasized sample, f is the physical frequency, α is the pre-emphasis coefficient, X_m is the MFCC vector element, x_n is the log Mel output, $\beta = 1$ for $m = 0$, else $\beta = 2$, for $m = 1$ to $(M-1)$, E is the number of Mel filters.

B. MFCC Hardware Optimization

To enable low power MFCC module, three hardware algorithm co-optimizations are proposed at the pre-emphasis, FFT and Mel filtering sub-modules respectively. Finally, piecewise linear (PWL) approximation method is employed to efficiently implement the logarithmic operation.

The pre-emphasis sub-module uses a hardware friendly coefficients, $\alpha = \frac{2^n-1}{2^n}$. As α typically ranges from 0.9 to 1, $n \geq 4$. Compared to the conventional design, this approach only requires shift and addition operation, as shown in Figure 4, hence having a 30% reduction in power.

The Hanning windowing sub-module simply consists of a look-up table (LUT) to store Hanning coefficients and a multiplier to perform windowing computation.



(a) Conventional (b) Optimized (c) Power reduction

Fig. 4. (a) Conventional and (b) Optimized pre-emphasis sub-module, as well as (c) power reduction after optimization.

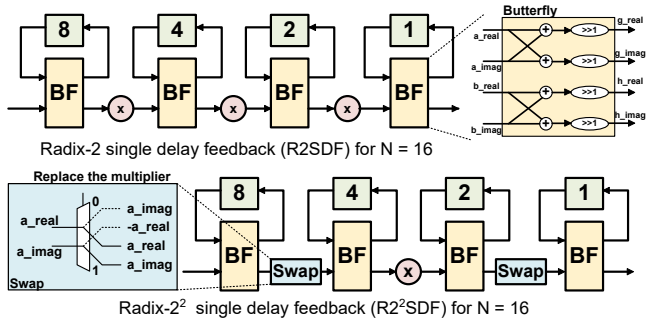
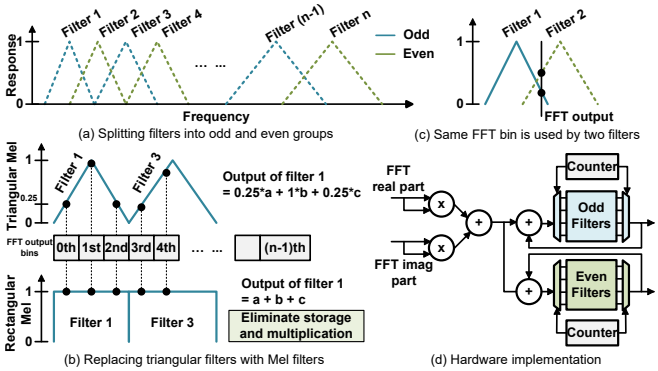


Fig. 5. R2SDF and R2^{2}SDF FFT hardware implementation.

For the FFT sub-module, radix-2^2 single delay feedback (R2^{2}SDF) FFT hardware implementation is chosen. Both R2^{2}SDF and conventional R2SDF hardware designs are derived from the re-expression of FFT computation using the Cooley–Tukey algorithm [13]. The re-expression leads to the reduction of FFT computation complexity and butterfly operation that can be implemented using few simple hardware components. The butterfly structure can be found in both R2SDF and R2^{2}SDF, as illustrated in Figure 5. However, the unique point of R2^{2}SDF is that it replaces the complex multipliers in between the butterfly structures because the FFT is re-expressed using a radix of 4 [14]. As a result, R2^{2}SDF utilizes lesser number of multipliers and adders as compared to the R2SDF implementation [14]. Therefore, the choice of R2^{2}SDF results in smaller and low power FFT sub-module.

To reduce the power of the Mel-filtering sub-module, triangular Mel filters are replaced by rectangular filters as shown in Figure 6(b). The rectangular filters of response of 1 not only save computation by avoiding the multiplication, but also eliminate the memory storage completely. Despite the



(a) Splitting filters into odd and even groups (c) Same FFT bin is used by two filters

(b) Replacing triangular filters with Mel filters (d) Hardware implementation

Fig. 6. (a) Splitting Mel filters into odd and even groups, (b) triangular Mel filters are replaced by rectangular filters to save memory and computation, (c) the same FFT bin is used by two Mel filters, and (d) hardware design of the Mel filtering sub-module.

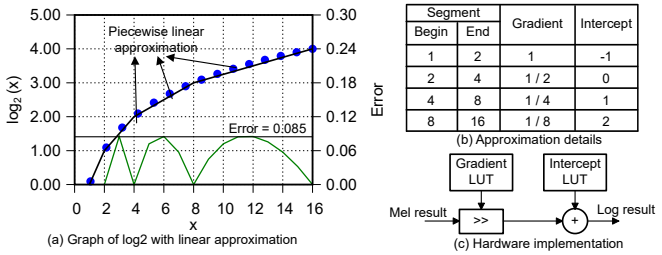


Fig. 7. (a) Approximating $\log_2$ using piecewise linear (PWL) equations, (b) approximation table, and (c) hardware implementation of $\log_2$.

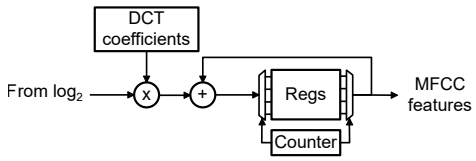


Fig. 8. DCT hardware implementation.

rectangular filters affecting the retained information, literature had showed that it has negligible effect on the classification accuracy [15]. In the hardware implementation, the sub-module receives the FFT complex output. It is then squared to obtain the magnitude. Each FFT bin magnitude is used by one odd and even numbered filter, as shown in Figures 6(a) and 6(c). Thus, the accumulation occurs in parallel and the result is stored in the respective registers, as shown in Figure 6(d).

Logarithm operation of base 2 ($\log_2$) is used in our proposed MFCC hardware. In general, there is no specific choice for the base of the logarithm operation. However, for simpler hardware implementation, $\log_2$ is chosen. As $\log_2$ is a non-linear function, it is approximated using piecewise linear (PWL) functions, as shown in Figure 7(a). Each segment spans from 2^n to 2^{n+1}. Thereafter, the gradient and intercept value of each segment are extracted, as shown in Figure 7(b). The gradient has an interesting property where the multiplication between the input and gradient can be simplified by using right shift operation. The values of right shift for all segments are encoded in a LUT. Similarly, the intercept values are also stored in a LUT table. Using the PWL approximation method, the maximum error is about 0.08. Unlike [7], its approximation always takes the floor of logarithmic result, which results in simple hardware implementation but large error. In our experiment, the large approximation error has an adverse effect on the application's accuracy.

The DCT sub-module requires a LUT, a multiplier, an adder and a set of registers, as depicted in Figure 8. The corresponding DCT coefficient is read out and multiplied with the $\log_2$ output. The result is then accumulated and stored temporarily in the register. This hardware design directly realizes the computation.

With the proposed optimization, the MFCC processing achieves a 1.5× and 1.2× reduction in number of multiplication and additions respectively, as shown in Figure 9. As a result, the computation power is also reduced by about 1.5×.

	Conventional		Proposed		Reduction	
	#mults	#adds	#mults	#adds	#mults	#adds
Pre-emphasis	N	N	--	N	100%	--
Windowing	N	--	N	--	--	--
FFT	$2N \log_2 N$	$6N \log_2 N$	$\frac{3N \log_2 N}{2}$	$5N \log_2 N$	25%	16%
Mel filtering	2N	2N	--	2N	100%	--
$\log_2$	--	--	--	N	--	--
DCT	MF	MF	MF	MF	--	--

N: FFT size F: number of Mel filters M: number of MFCC features Conventional FFT uses R2SDF

Fig. 9. MFCC workload breakdown before and after optimization.

TABLE I
THREE APPLICATION SCENARIOS USING MFCC FEATURES

Application	Keyword spotting	Bearing fault detection	Heart sound classification
Dataset	GSCD [16]	CWRU [17]	PhyisoNet 2016 [18]
Classifer output	10 keywords	13 faults	2 classes
Classifier[1]	LSTM (1-layer of 64 nodes + 1 FC)	CNN (2 CONVs + 1 FCs)	CNN (2 CONVs + 3 FCs)
Feature	MFCC	MFCC	MFCC
Accuracy (Ideal MFCC)	92.8%	96.3%	80.1%
Accuracy (Optimized MFCC)	92.6%	96.0%	78.6%

[1] CONV: convolutional layer, FC: fully-connected layer

C. Pipeline stages of proposed MFCC module

The proposed MFCC module processes an input signal frame and produces an output via three pipeline stages, as shown in Figure 10. Each signal frame consists of N samples. Each frame sample of n-bit wide serially enters the pipeline stages. The first pipeline stage produces the FFT result of the input frame. The second stage produces the Mel filtering result. The output data from the second stage is $2n$-bit wide to prevent data resolution loss. The third pipeline stage produces the final MFCC output of m-bit long. The number of clock cycles of the first and second stage is N. While, the third stage requires MF clock cycles, which is independent of FFT size, N. The operating frequency of the proposed hardware is related to the sampling frequency, FFT size and the number of clock cycles of all pipeline stages. For example, assume that an input signal is sampled at 8kHz and the FFT size, N =128, operating frequency of 16kHz is sufficient for real time MFCC processing.

III. Performance Evaluation

A. Impact on classification accuracy

The proposed MFCC module is verified in three application scenarios to evaluate the impact of quantization and approximated MFCC due to the optimizations described in Section II-B on classification accuracy, as shown in Table I. For the keyword spotting task, a long short term memory (LSTM) network is trained with the Google Speech Commands dataset [16]. For the bearing fault detection, a convolutional neural network (CNN) is trained using the CRWU bearing dataset [17]. Another CNN is trained to classify normal and abnormal heart sound from the dataset of the PhysioNet Computing in

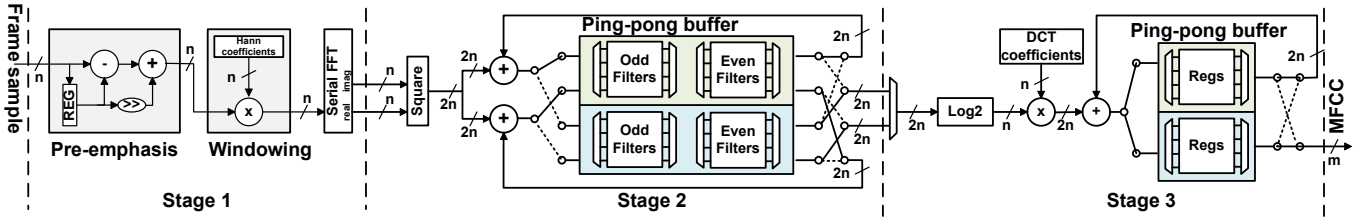


Fig. 10. Three pipeline stages of the proposed MFCC hardware.

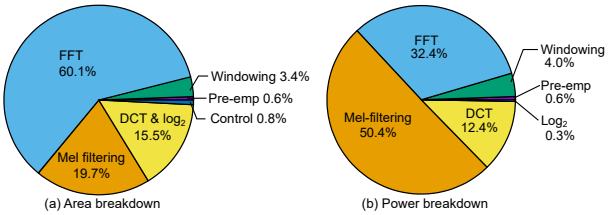


Fig. 11. (a) Area and (b) power breakdown of the proposed MFCC engine.

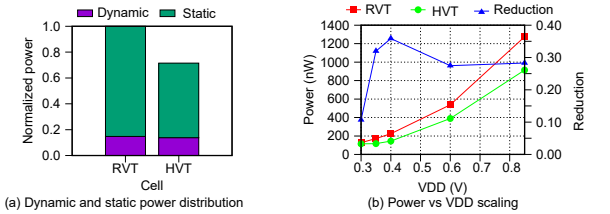


Fig. 12. (a) Dynamic and static power and (b) total power vs voltage, of the proposed MFCC hardware.

Cardiology Challenge 2016 [18]. All networks are trained and tested using both ideal (i.e. floating point in software) and the approximated MFCC. As compared to the pure software MFCC implementation, the approximated MFCC incurs negligible ($\leq$1.5%) accuracy drop. This confirms the suitability of the proposed design for hardware-based edge AI applications.

B. Power and Area Efficiency

To demonstrate the benefits of the proposed optimization, a MFCC engine, which is targeted for keyword spotting task reported in Section III-A, is implemented. Our MFCC hardware is configured to have FFT size of 128, 16 Mel filters, 11 MFCC features and operating frequency of 16 kHz. At this frequency, the engine can operate down to 0.3V using 40nm CMOS process.

Figure 11(a) shows area breakdown of the proposed MFCC engine. The FFT sub-module takes up most of the silicon area (60%), followed by Mel-filtering (20%) and DCT (16%) sub-modules, mainly due to registers to store intermediate results. Figure 11(b) shows the power consumption of the proposed hardware. The Mel-filtering sub-module consumes the most power (50%), followed by the FFT (32%) and DCT (12%).

Figure 12(a) shows the static and dynamic power of the proposed MFCC module. Since the proposed MFCC module runs at very low frequency, the static power becomes dominant. To further reduce the power consumption, our design is re-synthesized using the high threshold voltage (V_T) standard cells (HVT cells), and the total power is further reduced by about 30%. Figure 12(b) shows the power of our design that is simulated at different V_{DD} using Cadence Virtuoso AMS.

TABLE II
COMPARISON TABLE

	ISSCC 2020 [7]	JSSC 2020 [9]	Access 2020 [11]	ISVLSI 2020 [19]	This work
Tech, nm	28	65	180	180	40
Feature	MFCC	MFCC	MFCC	MFCC	MFCC
Type	Digital	Digital	Mixed	Mixed	Digital
Voltage, V	0.4	0.6	1.8	1.8	0.3
Freq, kHz	40	250	–	–	16
Area, mm^2	0.11[1]	–	1.0	2.16	0.08
Power, nW	340[1]	7500[1]	21400	33000	128
N. area[2] mm^2	0.22	–	0.05	0.11	0.08
N. power[2] nW	694	2840	1067	1630	128

[1] Only MFCC feature extraction hardware
[2] "N." refers to "normalized", the area and power data is normalized to 40nm

At 0.3V, the proposed design attains the lowest power among other V_{DD} despite the low gain in power reduction.

Table II compares the proposed MFCC engine with existing works. For a fair comparison, the area and power data are normalized to 40nm. Both area and power are multiplied with the same scaling factor of $\frac{1}{s^2}$, where $s = \frac{40}{\text{original node}}$, as suggested in [20] and [21]. The post-synthesis power estimated using Cadence Virtuoso AMS simulation is reported for our proposed MFCC engine. It achieves 128nW of power, which is at least 5$\times$ lower power compared to all digital MFCC hardware designs [7], [9]. Also, the proposed MFCC hardware has 2.75$\times$ smaller area when compared to the digital MFCC hardwares listed in Table II. Furthermore, the proposed MFCC engine derives at least about 8$\times$ power reduction at comparable silicon area as the mixed-signal MFCC hardware reported in [11] and [19]. The proposed MFCC engine therefore demonstrates great potential to be deployed on low power and low cost edge devices.

IV. Conclusion

This work proposes a low power MFCC hardware by adopting various optimizations at all MFCC processing steps. Our optimized MFCC processing is tested with three application scenarios showing negligible or no accuracy drop. Our implemented MFCC hardware is configured for a keyword spotting task. The resulting MFCC hardware achieves 128nW at 0.3V based on a 40nm technology node. The hardware performance achieves 5$\times$ power reduction as compared to the relevant digital MFCC implementation.

Acknowledgment

We thank the Programmatic grant no. A1687b0033, Singapore RIE 2020, AME domain.

REFERENCES

[1] M. Deng *et al.*, "Heart sound classification based on improved MFCC features and convolutional recurrent neural networks," *Neural Networks*, vol. 130, pp. 22–32, oct 2020.

[2] Y. Lu, W. Shan, and J. Xu, "A Depthwise Separable Convolution Neural Network for Small-footprint Keyword Spotting Using Approximate MAC Unit and Streaming Convolution Reuse," in *2019 IEEE Asia Pacific Conference on Circuits and Systems*, nov 2019, pp. 309–312.

[3] Q. Jiang *et al.*, "Bearing Fault Classification Based on Convolutional Neural Network in Noise Environment," *IEEE Access*, vol. 7, pp. 69 795–69 807, 2019.

[4] X. Huang *et al.*, *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*, 1st ed. Prentice Hall PTR, 2001.

[5] M. P. Norton *et al.*, "Fundamentals of Noise and Vibration Analysis for Engineers," *Fundamentals of Noise and Vibration Analysis for Engineers*, sep 2003.

[6] S. B. Davis *et al.*, "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.

[7] 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," *IEEE Journal of Solid-State Circuits*, pp. 1–1, oct 2020.

[8] Y. S. Chong *et al.*, "A 2.5 uw kws engine with pruned lstm and embedded mfcc for iot applications," *IEEE Transactions on Circuits and Systems II: Express Briefs*, pp. 1–1, 2021.

[9] 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, apr 2020.

[10] N. V. Vu *et al.*, "Implementation of the MFCC front-end for low-cost speech recognition systems," *2010 IEEE International Symposium on Circuits and Systems*, pp. 2334–2337, 2010.

[11] Q. Li *et al.*, "MSP-MFCC: Energy-Efficient MFCC Feature Extraction Method with Mixed-Signal Processing Architecture for Wearable Speech Recognition Applications," *IEEE Access*, vol. 8, pp. 48 720–48 730, 2020.

[12] K. S. Rao *et al.*, "Appendix A MFCC Features," in *Speech Recognition Using Articulatory and Excitation Source Features*, 2017, pp. 85–88.

[13] J. W. Cooley *et al.*, "An algorithm for the machine calculation of complex Fourier series," *Mathematics of Computation*, vol. 19, no. 90, pp. 297–301, may 1965.

[14] S. He *et al.*, "New approach to pipeline FFT processor," *IEEE Symposium on Parallel and Distributed Processing*, pp. 766–770, 1996.

[15] W. Han *et al.*, "An efficient MFCC extraction method in speech recognition," *IEEE International Symposium on Circuits and Systems*, pp. 145–148, 2006.

[16] P. Warden, "Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition," *arXiv*, apr 2018.

[17] "Bearing Data Center." [Online]. Available: https://engineering.case.edu/bearingdatacenter

[18] C. Liu *et al.*, "An open access database for the evaluation of heart sound algorithms," *Physiological Measurement*, vol. 37, no. 12, p. 2181, nov 2016.

[19] Y. Zhang *et al.*, "Optimization and evaluation of energy-efficient mixed-signal MFCC feature extraction architecture," *Proceedings of IEEE Computer Society Annual Symposium on VLSI*, vol. 2020-July, pp. 506–511, jul 2020.

[20] M. T. Bohr and I. A. Young, "CMOS Scaling Trends and beyond," *IEEE Micro*, vol. 37, no. 6, pp. 20–29, nov 2017.

[21] G. G. Shahidi, "Chip Power Scaling in Recent CMOS Technology Nodes," *IEEE Access*, vol. 7, pp. 851–856, 2019.