An Area-Efficient Ultra-Low-Power Time-Domain Feature Extractor for Edge Keyword Spotting

Qinyu Chen*, Yaoxing Chang*, Kwantae Kim*, Chang Gao† and Shih-Chii Liu*
*Institute of Neuroinformatics, University of Zurich and ETH Zurich, Switzerland
†Department of Microelectronics, Delft University of Technology, Netherlands

2023 IEEE International Symposium on Circuits and Systems (ISCAS) | 978-1-6654-5109-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/ISCAS46773.2023.10181602

***Abstract*—Keyword spotting (KWS) is an important task on edge low-power audio devices. A typical edge KWS system consists of a front-end feature extractor which outputs mel-scale frequency cepstral coefficients (MFCC) features followed by a back-end neural network classifier. KWS edge designs aim for the best power-performance-area metrics. This work proposes an area-efficient ultra-low-power time-domain infinite impulse response (IIR) filter-based feature extractor for a KWS system. It uses a serial architecture, and the architecture is further optimized for a low-cost computing structure and mixed-precision bit selection of the IIR coefficients while maintaining good KWS accuracy. Using a 65 nm process technology and a back-end neural network classifier, this simulated feature extractor has an area of 0.02 mm^2 and achieves 3.3 μW @ 1.2 V, and achieves 92.5% accuracy on a 10-keyword, 12-class KWS task using the GSCD dataset.**

***Index Terms*—Keyword spotting (KWS), infinite impulse response (IIR), hardware acceleration, long short-term memory.**

I. Introduction

KEYWORD spotting (KWS) is an important task to reduce power consumption on consumer devices and smart assistants such as Apple Siri. It acts as a trigger for the system to perform the higher-computation and energy-consumption speech recognition tasks. Ideally, an edge KWS system should be always-on and for battery-powered devices, an ultra-low power design is needed.

A typical KWS design consists of a front-end feature extractor and a back-end neural network classifier [1]. Deep neural networks (DNNs) such as Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) are usually used as the back-end classifier [2]–[4]. In recent years, a few dedicated KWS FPGA and ASIC designs have been designed with focus on reducing power consumption [5]–[12].

In addition to the back-end, the front-end feature extractor is another crucial module in speech recognition systems. The recent development of edge audio systems has heightened the need for area-power efficient feature extractors. Audio features can be categorized into two main types: 1) frequency-domain features such as the mel-scale frequency cepstral coefficients (MFCC) [8], [13], [14], and 2) time-domain features, such as ring-oscillator-based filters [7] or g_mC filters [15], [16]. MFCC features are typically used in state-of-the-art KWS digital systems [6], [8], [13], [17]–[19]. However, processing chains of MFCC mandates a large power consumption including pre-emphasis, windowing, fast Fourier transform (FFT), Mel-filtering, logarithmic computation, and discrete cosine transform (DCT). For example, in the 22nm 10.8μW/15.1μW dual computing modes KWS system [17], the MFCC feature extractor took up nearly one-third of the power consumption of the whole prototype system.

Corresponding authors: Shih-Chii Liu and Chang Gao (shih@ini.uzh.ch, chang.gao@tudelft.nl)

This work was partially supported by the Swiss National Science Foundation CA-DNNEdge project (208227) and Bridge project VIPS (181010).

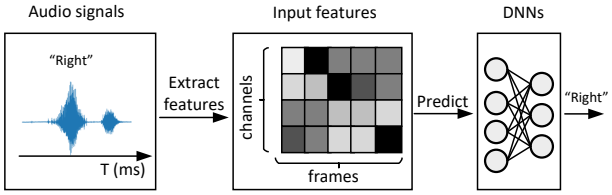

Fig. 1. Typical processing pipeline in a KWS system.

Infinite impulse response (IIR) filters are a class of widely-used recursive filters and have relatively simple computing chains thereby facilitating low memory and logic area costs [20]. In this work, we propose an area-efficient ultra-low-power digital serial IIR filter-based feature extractor as a front-end to a KWS system. The low-cost computing structure and mixed-precision implementation demonstrates a superior area-power efficiency. Designed in a 65 nm CMOS process, the feature extractor shows a 1.6x-to-54x lower power consumption than other designs while recording a 92.5% state-of-the-art KWS accuracy on the Google Speech Command Dataset (GSCD) [21].

II. Infinite Impulse Response Filter

IIR filters are defined by a set of coefficients of a rational transfer function, for which a stability criteria must be also satisfied. By doing a z-transform, the transfer function of the IIR filter is as follows:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{i=0}^{M-1} a_i z^{-i}}{1 + \sum_{i=1}^{N-1} b_i z^{-i}} \tag{1}$$

where a_i and b_i are coefficients in the numerator and denominator, respectively. M and N denote the number of poles and zeros in the filter function.

The most common IIR filter model is based on the decomposition of a higher-order filter into a cascade of second-order sections to ensure stability [22]. In this way, we can describe the transfer function of a higher order IIR filter as follows:

$$H(z) = \prod_{i=i}^{L} \frac{a_{0_i} + a_{1_i} z^{-1} + a_{2_i} z^{-2}}{1 + b_{1_i} z^{-1} + b_{2_i} z^{-2}} \tag{2}$$

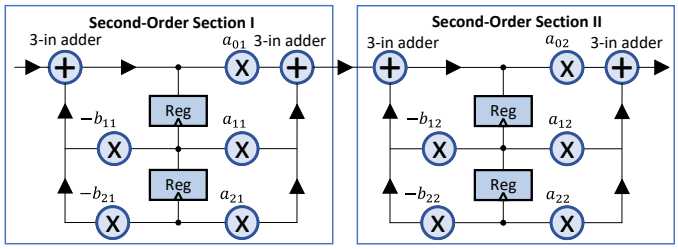

Fig. 2. Basic architecture of a 4^{th} order IIR band-pass filter.

TABLE I
COEFFICIENTS OF 16 IIR BAND-PASS FILTERS IN SECOND-ORDER SECTIONS REPRESENTATION

Ch	a_{01}	a_{11}	a_{21}	b_{11}	b_{21}	a_{02}	a_{12}	a_{22}	b_{12}	b_{22}	CF (Hz)
0	5.4-e04	1.1e-03	5.5e-04	-1.93	0.96	1	-2	1	-1.96	0.97	182
1	1.2e-03	2.4e-03	1.2e-03	-1.89	0.94	1	-2	1	-1.92	0.95	272
2	0.0022	0.0044	0.0022	-1.82	0.93	1	-2	1	-1.88	0.94	372
3	0.0036	0.0072	0.0036	-1.74	0.91	1	-2	1	-1.82	0.92	482
4	0.0056	0.011	0.0056	-1.63	0.88	1	-2	1	-1.74	0.90	603
5	0.0081	0.016	0.0081	-1.49	0.86	1	-2	1	-1.64	0.89	736
6	0.011	0.023	0.011	-1.31	0.84	1	-2	1	-1.53	0.86	884
7	0.016	0.031	0.016	-1.10	0.81	1	-2	1	-1.38	0.84	1046
8	0.021	0.042	0.021	-0.85	0.78	1	-2	1	-1.21	0.81	1225
9	0.027	0.054	0.027	-0.56	0.76	1	-2	1	-0.99	0.78	1422
10	0.035	0.070	0.035	-0.23	0.73	1	-2	1	-0.75	0.75	1639
11	0.044	0.089	0.044	0.14	0.70	1	-2	1	-0.46	0.71	1879
12	0.056	-0.11	0.056	-0.15	0.67	1	2	1	0.53	0.68	2143
13	0.069	-0.14	0.069	0.18	0.61	1	2	1	0.92	0.66	2435
14	0.085	-0.17	0.085	0.51	0.54	1	2	1	1.29	0.67	2756
15	0.10	-0.21	0.10	0.78	0.44	1	2	1	1.64	0.74	3110

where L is the number of second-order sections. Fig. 2 shows a basic architecture of a 4^{th} order IIR band-pass filter implemented as a cascade of two second-order sections (L=2). In this work, we design a 16-channel IIR-based feature extractor with 16 IIR filters. We use *signal.butter* function in the *Scipy* library to design *Butterworth* IIR band-pass filters. We chose *Butterworth* filter topology as it has a flat passband response, i.e., without passband ripple as it exists in Chebyshev/Elliptic filters, thereby leading to a easier and simpler design. The coefficients and central frequencies of the filter channels are given in Table I.

III. LOW-POWER AND COMPACT SERIAL IIR-BASED FEATURE EXTRACTOR

A. Serial Architecture and Computational Flow

Fig. 3 shows the top view of the proposed IIR-based feature extractor. It consists of a computing unit which includes a cascade of second-order section modules, a post-processing module, buffers and a control unit. The input audio signals are processed sequentially by the two second-order section modules and the post-processing module to generate the 16-dimensional feature vectors. The serial architecture allows the reuse of the computing units across the 16 channels and thereby, significantly reducing both the area and leakage power of the design. Fig. 4 shows the computation flow of the serial IIR-based feature extractor. The audio samples are

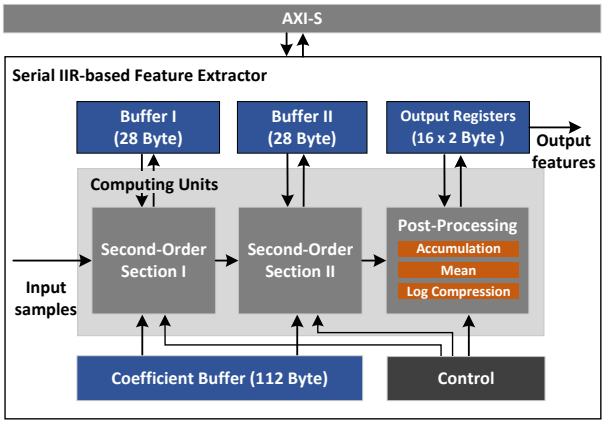

Fig. 3. Hardware architecture of a serial IIR-based feature extractor (within the outlined box).

partitioned into 16 ms frames without overlap. Each input frame contains 128 samples at the sampling rate of 8 kHz. Each audio sample is processed by the 16 IIR band-pass filters in a sequential manner by the same computation units. The computation units are divided into three pipeline stages to reduce the combinational logic area and power consumption. After each of the 128 audio samples is filtered and the outputs accumulated in the output registers, their values are divided by 128 (realized by a bit right shift) to obtain the mean of a frame, then it is followed by a log compression. Finally, the values in the output registers are streamed out as a 16-dimensional feature vector.

B. Low-Cost Computing Structure of the IIR Band-pass Filter

According to Fig. 2, which demonstrates the basic architecture of a 4^{th} order IIR band-pass filter, 10 multipliers and 8 adders are required. If we look closer at the coefficients a and b across the 16 IIR filters as shown in Table I, we see hardware-friendly properties such as symmetries and constant value representations, e.g.:

$$a_{11} = 2a_{01} = 2a_{21}, \tag{3}$$

$$|a_{12}| = 2a_{02} = 2a_{22} = 2, \tag{4}$$

Shown in Fig. 5, a low-complexity architecture for the 4^{th}-order IIR filter is designed by exploiting these properties. The number of multipliers are reduced to 5 by the replacement with shift operations and combining like terms, thus reducing both the power consumption and area.

C. Mixed-Precision Selection of Filter Coefficients

Due to potential numerical stability issues, a fixed-point analysis is required during the IIR filter implementation. The resource and power reduction strongly depends on the coefficient values of the filters. In this case, the selection of coefficient precision is essential. Previous works always adopt the unified bit precision for coefficients and focus on the goodness of fitting degree during the fixed-point analysis. In fact, thanks to the error resilience of neural networks, we

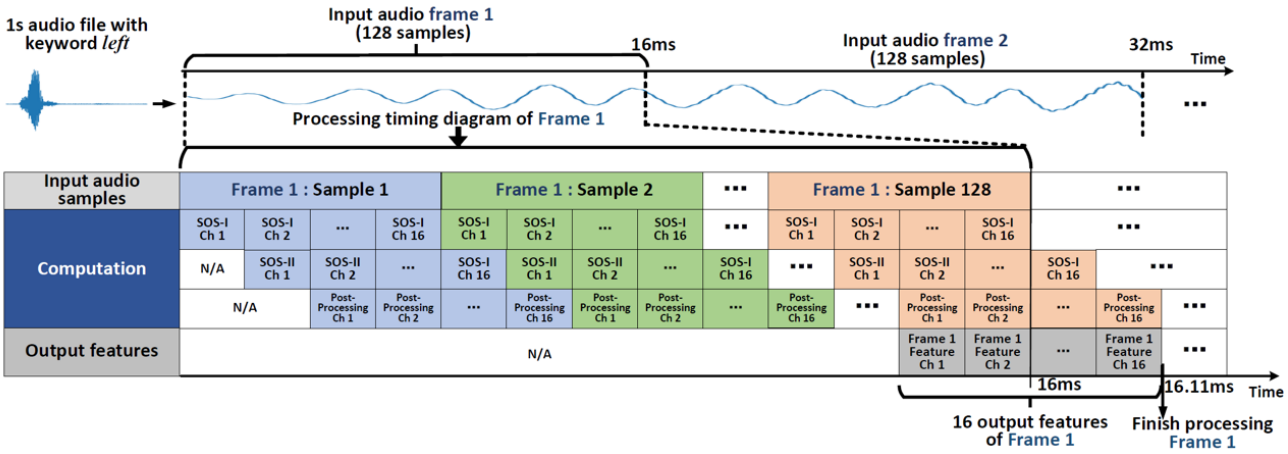

Fig. 4. Timing diagram of serial IIR-based feature extractor with 16 channels. Each audio frame has 128 samples (16ms frame length, 8KHz sampling rate, no frame overlap). Operations in horizontal and vertical directions are executed sequentially and concurrently, respectively.

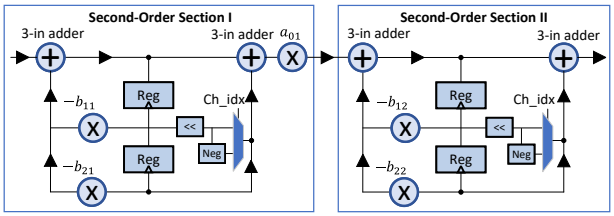

Fig. 5. Improved architecture of a 4^{th} order IIR band-pass filter. (Ch_idx denotes the channel index).

can select an aggressive bit-width selection, in other words, the network accuracy is used as the measurement standard. Through experiments, we find that the dynamic ranges for coefficients a and b across the feature extractor are different, and the impact of the bit precision on the KWS accuracy also varies. The independent bit precision selection allows for further reduction of power and hardware resources. Algorithm 1 shows the selection of the bit precision for coefficients a and b. The integer bits are first determined by the maximum value of a and b to prevent overflow, separately. The fraction bits are then reduced from the baseline and the network accuracy is again quantified. The selection program stops once the accuracy is lower than the predefined threshold, Acc_{th}, which is also the expected KWS accuracy.

Algorithm 1 Bit Selection of IIR Filter Coefficients

Input:
$LSTM$, the network used to evaluate the features after quantization. $\{Q_f^a,\ Q_f^b\}$, the predefined fraction bits of coefficients a and b; F_q, quantized features; Acc, the validation accuracy; Acc_{th}, the predefined accuracy threshold.

Output:
$\{q_i^a,\ q_f^a,\ q_i^b,\ q_f^b\}$, the integer and fraction bits of coefficient a and b;

q_i^a = len(bin(max(a))), and q_i^b =len(bin(max(b))
for l, r in $\{\ \{\ Q_f^a,\ Q_f^b\},\ \{q_f^a,\ q_f^b\}\ \}$ **do**
  for $i = 0; i \leq l; i++$ **do**
    r = l - i. // Update the List R
    Acc = Evaluate($LSTM$, F_q).
    if $Acc \leq Acc_{th}$ **then**
      BreakLoop()
    end if
  end for
end for

IV. Experimental Results

We use a single-layer LSTM model of 64 hidden nodes to validate our design. The network is trained to predict 12 classes, including 10 keywords: *down*, *go*, *left*, *no*, *off*, *on*, *right*, *stop*, *up*, *yes*, together with *silence* and *unknown* classes using GSCD. The audio clips are downsampled from 16 kHz to 8 kHz, to reduce computation without any accuracy loss. We first normalize the audio files, and then estimate the impact of the different precision selections of coefficients a and b on the network accuracy over multiple runs. For the normalization, each audio file is 1) subtracted by the mean, then 2) clipped and 3) divided by the five times of the standard deviation. The normalization by itself brings around 1% accuracy improvement. Fig. 6 (a) shows the relationship between the accuracy and bit precision of coefficient b, while bit precision of coefficient a is set to full precision (32-bit floating point). The accuracy is maintained at around 92.5% when the bit precision of b, i.e. $b_b \geq 8$. Lower precision for b will result in numerical instability of the IIR filter output. Thus, the generated features led to a poor accuracy of 8.3% from the network. Fixing $b_b = 8$, we then investigated how the accuracy is impacted by the bit precision of a. As shown in Fig. 6 (b), the accuracy begins to gradually drop when the bit precision is a, i.e. $b_a < 12$. For the final classification results, we set $b_b = 8$ and $b_a = 12$, to guarantee no accuracy loss compared to the baseline.

Our serial IIR-based feature extractor is implemented at Register Transfer Level (RTL) using System Verilog, synthesized by Synopsys Design Compiler, and designed with

TABLE II
COMPARISON WITH PREVIOUS WORKS

Metric	JSSC'20 [8]	TCAS-II'22 [13]	TCAS-I'20 [17]	VLSI'19 [23]	JSSC'22 [7]	This work
Technology (nm)	65	40	22	65	65	65
Sample rate (kHz)	16	8	16	4-16	-	8
Frequency (kHz)	250	400	250	250	0.11-10.4	128
Voltage (V)	0.6	0.6	0.6	0.6	0.5	1.2
Precision (bits)	-	-	8@Input 16@Output	10@Input 4,8@Output	-	12@Input, 12@Output b_a=12, b_b=8
Latency (ms)	48	16	40	16	-	16.11
Feature extractor type	Digital MFCC	Digital MFCC	Digital MFCC	Digital MFCC	Analog ring-oscillator	Digital IIR-based
Power (μW)	7	**0.67**	2.8	7.14	9.3	3.3
Area (mm^2)	-	-	<0.1	-	1.6	**0.02**
#Keywords / #Classes	10/-	10/12	10/12	10/12	10/12	10/12
Model[1]	LSTM	LSTM	BCNN	LSTM	GRU	LSTM
Accuracy[2]	90.87% ^{m}	92.40% ^{s}	87.90%^{s}	90.87% ^{m}	86.03% ^{m}	**92.50%** ^{s}

[1] All LSTMs have one 64-unit hidden layer structure; BCNN – binary convolutional neural network; GRU – gated recurrent unit network.
[2] m denotes chip measurement results, s denotes simulation results.

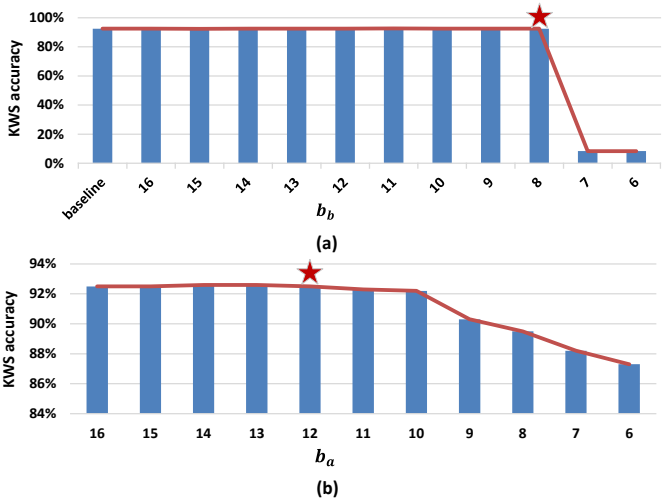

Fig. 6. Impact of bit precision on KWS accuracy: (a) KWS accuracy versus b_b ($b_a = 32$ floating point, baseline), (b) KWS accuracy versus b_a ($b_b = 8$).

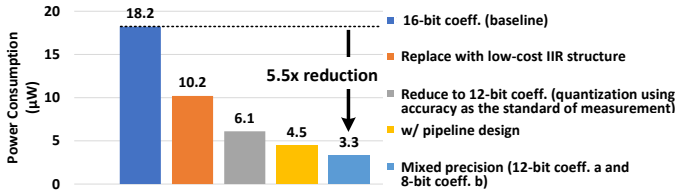

Fig. 7. Power reduction with each optimization operation.

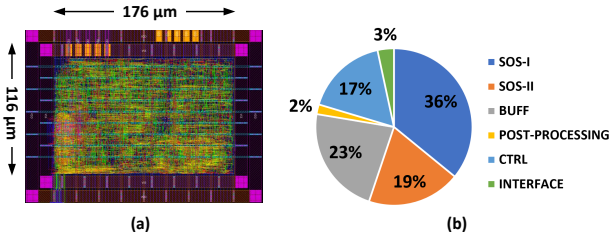

Fig. 8. (a) Layout and (b) power breakdown of the IIR-based feature extractor.

Cadence Innovus for place and route using CMOS 65nm process. The pre-layout and post-layout functionalities are verified using ModelSim, and stimuli-based power consumption is analyzed from post-layout results. To offer real-time processing capability, the proposed design runs at a system clock of 128 kHz and a supply voltage of 1.2 V. Fig. 7 shows a final 5.5x power reduction with the breakdown across all the different optimization steps. The 16-channel IIR filter layout has an area of 0.02 mm^2 as shown in Fig. 8(a) and the power breakdown of the different blocks is shown in Fig. 8(b). The second-order section modules are the most power-consuming blocks, followed by the buffers.

Table II compares our proposed digital IIR-based feature extractor with the state-of-the-art extractors including both digital implementations [8], [13], [17], [23] and analog implementations [7]. Unlike MFCC-based designs, the hardware resources and power consumption of the proposed IIR-based feature extractor is independent of the frame length (referred as FFT size in MFCC). Our work achieves a competitive accuracy of 92.5% comparable to other works reporting accuracies on a 10-keyword, 12-class KWS task using the GSCD dataset. Simulations show that this feature extractor design consumes only 3.3 μW @ 1.2 V.

V. CONCLUSION

This work proposes an area-efficient ultra-low-power time-domain IIR-based feature extractor for edge keyword spotting. The implementation uses a serial architecture and is further optimized with a low-cost computing structure and mixed-precision selection to significantly reduce the power consumption while maintaining high KWS accuracy. It has an area of 0.02 mm^2 and consumes only 3.3 μW @ 1.2 V. The power of the entire KWS design including the IIR-based feature extractor and a network with an energy-efficient architecture [24] will be reported in future work.

REFERENCES

[1] G. Chen, C. Parada, and G. Heigold, "Small-footprint keyword spotting using deep neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2014, pp. 4087–4091.

[2] A. B. Nassif, I. Shahin, I. Attili, M. Azzeh, and K. Shaalan, "Speech recognition using deep neural networks: A systematic review," *IEEE access*, vol. 7, pp. 19 143–19 165, 2019.

[3] A. Coucke, M. Chlieh, T. Gisselbrecht, D. Leroy, M. Poumeyrol, and T. Lavril, "Efficient keyword spotting using dilated convolutions and gating," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2019, pp. 6351–6355.

[4] R. Tang and J. Lin, "Deep residual learning for small-footprint keyword spotting," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2018, pp. 5484–5488.

[5] L. Guo, P. Lin, L. Guo, and B. Liu, "Implementation of a CRNN-based low-power keyword recognition system on fpga," in *IEEE 14th International Conference on ASIC (ASICON)*, 2021, pp. 1–4.

[6] 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 J. Solid-State Circuits*, vol. 56, no. 1, pp. 151–164, Jan. 2021.

[7] K. Kim, C. Gao, R. Graça, I. Kiselev, H.-J. Yoo, T. Delbruck, and S.-C. Liu, "A 23-μW keyword spotting IC with ring-oscillator-based time-domain feature extraction," *IEEE J. Solid-State Circuits*, vol. 57, no. 11, pp. 3298–3311, 2022.

[8] J. S. P. Giraldo, S. Lauwereins, K. Badami, and M. Verhelst, "Vocell: A 65-nm speech-triggered wake-up SoC for 10-μW keyword spotting and speaker verification," *IEEE J. Solid-State Circuits*, vol. 55, no. 4, pp. 868–878, 2020.

[9] D. Wang, S. J. Kim, M. Yang, A. A. Lazar, and M. Seok, "A background-noise and process-variation-tolerant 109 nW acoustic feature extractor based on spike-domain divisive-energy normalization for an always-on keyword spotting device," in *IEEE International Solid- State Circuits Conference (ISSCC)*, vol. 64, Feb. 2021, pp. 160–162.

[10] J. Giraldo and M. Verhelst, "Laika: A 5μW programmable LSTM accelerator for always-on keyword spotting in 65nm CMOS," in *ESSCIRC 2018 - IEEE 44th European Solid State Circuits Conference (ESSCIRC)*, 2018, pp. 166–169.

[11] C. Gao, T. Delbruck, and S.-C. Liu, "Spartus: A 9.4 TOp/s FPGA-Based LSTM Accelerator Exploiting Spatio-Temporal Sparsity," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–15, 2022.

[12] Q. Chen, Y. Huang, R. Sun, W. Song, Z. Lu, Y. Fu, and L. Li, "An efficient accelerator for multiple convolutions from the sparsity perspective," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 28, no. 6, pp. 1540–1544, 2020.

[13] Y. S. Chong, W. L. Goh, V. P. Nambiar, and A. T. Do, "A 2.5 μW KWS engine with pruned LSTM and embedded MFCC for IoT applications," *IEEE Trans. Circuits Syst. II, Express Briefs*, vol. 69, no. 3, pp. 1662–1666, 2022.

[14] B. U. Pedroni, S. Sheik, H. Mostafa, S. Paul, C. Augustine, and G. Cauwenberghs, "Small-footprint spiking neural networks for power-efficient keyword spotting," in *IEEE Biomedical Circuits and Systems Conference (BioCAS)*, 2018, pp. 1–4.

[15] S.-C. Liu, A. van Schaik, B. A. Minch, and T. Delbruck, "Asynchronous binaural spatial audition sensor with $2 \times 64 \times 4$ channel output," *IEEE Trans. Biomed. Circuits Syst.*, vol. 8, no. 4, pp. 453–464, 2014.

[16] K. Badami, S. Lauwereins, W. Meert, and M. Verhelst, "24.2 Context-aware hierarchical information-sensing in a 6W 90nm CMOS voice activity detector," in *IEEE Int. Solid- State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2015, pp. 1–3.

[17] B. Liu *et al.*, "A 22nm, 10.8μ W/15.1μW dual computing modes high power-performance-area efficiency domained background noise aware keyword- spotting processor," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 12, pp. 4733–4746, 2020.

[18] P. P. Bernardo, C. Gerum, A. Frischknecht, K. Lübeck, and O. Bringmann, "Ultratrail: A configurable ultralow-power tc-resnet ai accelerator for efficient keyword spotting," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 11, pp. 4240–4251, 2020.

[19] Y.-H. Chiang, T.-S. Chang, and S. J. Jou, "A 14 μJ/decision keyword spotting accelerator with In-SRAM-Computing and on chip learning for customization," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, pp. 1–9, 2022.

[20] Y. Chen, X. Zhang, Y. Lian, R. Manohar, and Y. Tsividis, "A continuous-time digital IIR filter with signal-derived timing and fully agile power consumption," *IEEE J. Solid-State Circuits*, vol. 53, no. 2, pp. 418–430, 2017.

[21] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," *arXiv preprint arXiv:1804.03209*, 2018.

[22] R. Garcia, A. Volkova, M. Kumm, A. Goldsztejn, and J. Kühle, "Hardware-aware design of multiplierless second-order iir filters with minimum adders," *IEEE Transactions on Signal Processing*, vol. 70, pp. 1673–1686, 2022.

[23] J. S. P. Giraldo, S. Lauwereins, K. Badami, H. Van Hamme, and M. Verhelst, "18μW SoC for near-microphone keyword spotting and speaker verification," in *Proc. IEEE Symp. VLSI Circuits*, 2019, pp. C52–C53.

[24] S.-C. Liu, C. Gao, K. Kim, and T. Delbruck, "Energy-efficient activity-driven computing architectures for edge intelligence," in *2022 International Electron Devices Meeting (IEDM)*, 2022, pp. 21–2.