# Distributed, Intelligent Audio Sensing Enabled by Low-Power Integrated Technologies

by

Minchang Cho

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy (Electrical and Computer Engineering) in the University of Michigan 2020

Doctoral Committee:

Assistant Professor Hun Seok Kim, Co-Chair Professor Dennis M. Sylvester, Co-Chair Professor David T. Blaauw Assistant Professor Reetuparna Das Minchang Cho mincho@umich.edu ORCID iD: 0000-0002-4044-0276

© Minchang Cho 2020 All Rights Reserved To God and my family with love and gratitude

# TABLE OF CONTENTS

DEDICATION	ii
LIST OF FIGURES	v
LIST OF TABLES	viii
ABSTRACT	ix
CHAPTER	
I. Introduction	1
<ol> <li>Prior Art: Wireless Audio Sensor Motes</li> <li>Challenges of Achieving Millimeter-Scale Audio Sensing</li> <li>Dissertation Overview</li> </ol>	3 6 9
II. A Microwatt Power Audio Processing IC and 8Mb Streaming Flash Memory	11
<ul> <li>2.1 Introduction</li></ul>	11 12 14 18 22 25 27
III. A Picowatt Standby Power Neural Network Processor With Custom ISA and 7T SRAM for Sensor Applications	33
<ul> <li>3.1 Introduction</li></ul>	33 36 36 42

	3.2.3 Ultra-Low Leakage 7T SRAM Memory	44
3.3	Acoustic Object Detection Sensor System	45
3.4	Measurement Results	47
IV. An A	coustic Signal Processing Chip With Nanowatt Power	
Voice	Activity Detection	53
4.1	Introduction	53
4.2	VAD System Overview	56
4.3	Analog Front-End Implementation	58
	4.3.1 Overall Architecture	58
	4.3.2 Charge Pump and 10-V Level Shifter	59
	4.3.3 Low-Noise Amplifier and Programmable-Gain Am-	
	plifier	60
4.4	Digital Back-End Implementation	64
	4.4.1 Overall Architecture	64
	4.4.2 Binary DCT Mixer Sequence Generator	65
	4.4.3 IF Mixer and Extractor	67
	4.4.4 Neural Network Processor	68
4.5	Acoustic Signature Wakeup Detection	70
4.6	Measurement Results	71
4.7	Summary	78
V. Millir	neter-Scale Wireless Audio Sensor Node	81
5.1	The First Generation	81
5.2	The Second Generation	85
VI. Conc	lusion	93
IBLIOGRA	PHY	97

## LIST OF FIGURES

## Figure

1.1	The $\mu$ AMPS sensor node (2000)	3
1.2	(a) Audio sensor mote for acoustic scenes monitoring (2009) (b)	
	TinyEARS (2013)	4
1.3	InfiniTime $(2016)$ .	5
1.4	Energy consumption vs. compression ratio on (a) IMOTE2 platform	
	and (b) TELOS platform.	7
1.5	Sensor system lifetime vs. event activity: When the event of inter-	
	est occurs infrequently, the always-on detector would determine the	
	lifetime.	8
2.1	Miniaturized audio sensor challenges.	12
2.2	Overall architecture of audio processing IC	13
2.3	Analog front-end (AFE) and ADC circuits.	14
2.4	Measured input referred noise spectrum of AFE	15
2.5	Synchronous SAR ADC clock controller circuits and its operational	
	timing diagram.	16
2.6	Measured DNL, INL, and frequency spectrum of ADC	17
2.7	Proposed compression algorithm.	18
2.8	Polyphase Quadrature Filtering (PQF) process.	19
2.9	Compelxity reduction from the algorithm optimization	21
2.10	Polyphase quadrature filter architecture.	23
2.11	Compression engine architecture.	24
2.12	Operation principle of ping-pong streaming Flash	25
2.13	Chip die photo of (a) audio processing chip and (b) 8Mb NOR Flash	
	chip	27
2.14	(a) Acoustic measurement setup (b) power spectral density (PSD) of	
	AFE	28
2.15	Time domain signals for compressed and origial audio clip at (a)	
	normal speed speech and (b) slow speed speech	29
2.16	Measured compression ratio vs. sound quality trade-off	30
2.17	Measured power breakdown of (a) audio IC and (b) compression engine.	31
3.1	The power profile and average power consumption of NN processor	
	in (a) duty-cycled operation and (b) always-on operation	35

3.2	The microarchitecture of on-sensor NN processor.	42
3.3	7T HVT SRAM bitcell and layout.	44
3.4	Overall block diagram of acoustic sensor system for object detection.	46
3.5	Die photograph of acoustic sensor system with on-sensor NN processor.	47
3.6	Algorithm parameter sensitivity analysis	49
3.7	Neural network topology for acoustic object detection	51
3.8	Power breakdown of the proposed acoustic sensor system	52
4.1	Always-on voice activity detection as a wakeup mechanism. Ad- vanced processing is enabled upon voice activity detection to save	
	the overall power. $\ldots$	54
4.2	(a) VAD system architecture. (b) Operating principle of mixer-based	
	sequential frequency scanning	57
4.3	AFE block diagram with ULP and HP chains	59
4.4	10-V level shifter shifts up nominal VDD level to 10 V with periodic	
	refresh. Its waveforms are shown at right	60
4.5	(a) LNA circuit diagram. (b) $OTA_{MAIN1}$ . (c) $OTA_{AUX_N1}$ and their	
	bias implementation. $OTA_{AUX_P1}$ are implemented similarly with the	
	opposite type of transistors.	62
4.6	(a) ULP LNA output waveform with conventional DDA CMFB (red)	
	versus with proposed CMFB consisting of coupling capacitors and	
	DDA (black). (b) Its spectrum (simulated)	63
4.7	PGA circuit diagram.	63
4.8	Measured HP PGA output showing ULP-HP mode transition time.	
	By turning on fast settling switches for 30 ms, the settling time re-	
	duces from 6 s (black) to 100 ms (red). $\ldots$ $\ldots$ $\ldots$ $\ldots$	64
4.9	Digital backend architecture.	65
4.10	Binary DCT mixer sequence generator circuits	66
4.11	IF mixer and extractor circuits.	67
4.12	NN processor core architecture.	68
4.13	Measured power reduction from computational sprinting	69
4.14	(a) Inaudible acoustic signature wakeup detection. (b) Local MLS	
	signature generator using programmable LFSR. (c) Time-drift syn-	
	chronization scheme.	72
4.15	Die micrograph and system integration with MEMS microphone	73
4.16	(a) ULP PGA. (b) HP PGA input referred noise spectrum density	
	with different PGA gain settings (min, mid, and max gain)	73
4.17	Power spectral density referred to input (PSD RTI) for LNA, PGA,	
	and DSP. Two different applied tones (1 and 2 kHz) are mixed down	
	to 250 Hz in IF and extracted by DSP at two mixing frequencies each	
	(0.75  and  1.25  kHz for  1  kHz and  1.75  and  2.25  kHz for  2-kHz tone).	74
4.18	Measured power distribution of ULP mode (left) and HP mode (right).	75
4.19	ROC curves for ULP VAD mode with varying SNRs in the electrical	
	test (electrical connection to LNA, left) and SPLs in the acoustic test	
	(using speaker/integrated microphone in the sound chamber, right).	76

4.20	Acoustic testing setup. Proposed chip was integrated into the system-	
	on-board with a MEMS microphone and 3-D-printed lid and tested	
	in a sound chamber	76
4.21	Measurement results of acoustic signature wakeup detection with	
	MLS sequence of six stages, $N_{MLS} = 63$ , and $x_{MLS} = 1$ at various	
	SNRs, showing detection down to -10-dB SNR	77
4.22	Measurement results of acoustic signature wakeup detection with var-	
	ious LFSR stages, showing the tradeoff between the minimum re-	
	quired SNR versus worst case detection latency.	78
4.23	Measured waveform of the acoustic system that switches between	
	ULP and HP modes.	79
5.1	Overall block diagram of the first generation audio sensor node	82
5.2	The complete $6 \times 5 \times 4 \text{ mm}^3$ audio sensor node	84
5.3	Measured power profile of audio sensor node	85
5.4	Overall block diagram of the second generation audio sensor node	86
5.5	Self booting circuits using COTS components	87
5.6	Planar and side view diagram of system integration	89
5.7	Vertical view of (a) main processing stack and (b) storage stack	90
5.8	Audio sensor system operation cycle.	91
5.9	The complete $\phi 11$ mm × 3mm audio sensor node	92

# LIST OF TABLES

## <u>Table</u>

2.1	Measurement Summary of Analog Front-End	13
2.2	Measurement Summary of ADC	17
2.3	Compression Algorithm Comparison	22
2.4	Measurement Results of Compression Ratio	28
2.5	Summary of Audio Processing IC	31
2.6	Comparison of Embedded Flash ICs	32
3.1	Instruction Set of On-Sensor NN Processor	37
3.2	Comparison of Neural Network Processor	48
3.3	Measured Acoustic Object Detection Accuracy	50
4.1	Comparison of Feature Extractor	80
4.2	Comparison of Voice Activity Detector (VAD)	80
5.1	Comparison of Developed Audio Sensor Node	91

### ABSTRACT

Distributed audio sensing is promising to bring full bloom of a variety of applications to improve human life. However, despite of the continued efforts, the state-ofthe-art audio sensor node systems still remain at centimeter-scales in size, preventing true ubiquitous and unobtrusive deployment of them. Meanwhile, the silicon technology has been remarkably advanced, dictated by Moore's Law, and this enables a new opportunity to realize millimeter-scale of computing. In this dissertation, we explore a way to develop a millimeter-scale wireless audio sensor node system, by combining the integrated silicon technology, machine learning, and low-power circuit techniques.

This dissertation first presents an audio processing IC that performs audio acquisition and compression, consuming  $4.7\mu$ W. A new low-power compression algorithm and its accelerator consume only  $1.5\mu$ W to provide  $4-32\times$  real-time audio compression. Newly designed custom 8Mb embedded NOR Flash enables seamless audio streaming by a ping-pong buffering scheme.

Second, a picowatt-level standby power neural network processor is introduced for sensor applications. By combining custom instruction set architecture, compact SIMD microarchitecture, and ultra-low leakage SRAM memory, the processor consumes only 440pW of power at standby mode while achieves 400-GOPS/W of energy efficiency at active mode, which is suitable for modest neural network workloads on miniaturized sensor platforms. The proposed neural network processor is integrated in an acoustic object detection sensor system, and successfully demonstrates >90% of positive detection and <3% of false alarm for 5 acoustic targets detection. Next part of this dissertation is a voice and acoustic activity detector that uses a mixer-based architecture and ultra-low power neural network based classifier. By sequentially scanning 4 kHz of frequency bands and down-converting to below 500 Hz, feature extraction power consumption is reduced by  $4\times$ . The neural network processor employs computational sprinting, enabling  $12\times$  power reduction. The system also features inaudible acoustic signature detection for intentional remote silent wakeup of the system while re-using a subset of the same system components. The measurement results achieve 91.5%/90% speech/non-speech hit rates at 10 dB SNR with babble noise and 142 nW power consumption. Acoustic signature detection consumes 66 nW, successfully detecting a signature 10 dB below the noise level.

Finally, two generations of complete, fully functional energy-autonomous audio sensor nodes with millimeter-scale form factor are demonstrated. The systems use the proposed audio processing ICs and neural network processor integrated with a MEMS microphone, general-purpose microprocessor, 8Mb Flashes, RF transceiver with custom antenna, PV cells for energy harvesting and optical communication, and millimeter size batteries. The complete stand-alone systems achieve 1 hour (1st gen.) and 3.2 hours (2nd gen.) of continuous speech recording and energy-autonomous operation in room light.

The research in this dissertation is believed to pave a way towards distributed, intelligent audio sensing and computing.

## CHAPTER I

## Introduction

Recent advance of distributed sensing systems has made devices feasible that can sense their environment and perform actions based on the collected data. This new technology has attracted a lot of attention from both industry and research community, and has also opened up a myriad of civilian and military applications. Especially, the distributed systems that are wirelessly interconnected compose the wireless sensor network (WSN), which enables remote retrieving of video and audio streams, images, and scalar sensor data such as temperature, pressure or humidity. Moreover, the long-term collection of big data based on these networked sensors has gaining huge popularity for Internet of Things (IoT) applications, in accordance with the recent success of machine learning (ML) and artificial intelligence (AI). In this paradigm, every objects surrounding our daily lives will be wirelessly connected, gather information, and will identify, classify, infer, correlate, and fuse the information from heterogeneous sources, which is expected to bring benefits for the improved quality of our daily life.

To be benefited by massively distributed sensor networks, the size reduction of each sensor device is paramount. Minuscule size makes the sensors as easy-to-deploy and unobtrusive as possible, minimizing the disturbance to human activity. In addition, small form factor enables the placement of sensor devices in completely new locations where computing was absent before. In that vein, the 'smart dust' concept was proposed in [1, 2], envisioning the applications and usage scenario of a sensor node with the size of grain of salt, with primary focus on wireless communication and networking architecture. Since then, miniaturized wireless sensor nodes have been a popular topic of research in the cyber-physical systems community. More recently, M<sup>3</sup> (Michigan Micro-Mote) sensing platform was introduced in [3], with more focus on hardware design perspective. The platform features millimeter-scale modular architecture and various system-level techniques. With staircase stacking method of four bare-die ICs and one battery, the overall system only consumes 1.0 mm<sup>3</sup> of volume.

However, prior achievements of size shrinking have been only limited to sensor nodes with low dimensional sensing modalities, such as temperature [4], pressure [5,6], pH [7], and simple still cut images [8]. On the other hand, acoustic sensing is gaining more attention as it offers several advantages over other sensing modalities. The information carried by sound is comparable to that of video, yet requires much lower computational cost to be processed for the realization of highly resource constrained platforms. Sound is captured omni-directionally and intrinsically tolerant to light or obstacles, thus enables simultaneous multi-targets and/or events detection without careful positioning or adjustment. In addition, human voice is one of the most natural way to communicate with machines, avoiding the use of one's hands. Due to a variety of these merits, audio sensing and processing are widely adopted in extensive applications such as smart grid and home automation [9], ambient assisted living (e.q. patients support) [10], structural monitoring [11], biodiversity assessment [12], environmental monitoring for urban [13] and nature [14], surveillance [15, 16], localization [17], and voice user interfaces [18, 19]. Realizing audio sensing capability on miniaturized wireless sensor node will further provide enriched opportunities to broaden its applications, as the massively distributed audio sensor network will facilitate innovative information acquisition and processing to reshape interactions between



Figure 1.1: The  $\mu$ AMPS sensor node (2000).

people, environment, and devices.

#### 1.1 Prior Art: Wireless Audio Sensor Motes

The journey for the development of miniaturized wireless audio sensor node began in the late 1990's. In 1999, Rockwell Science Center and UCLA were engaged in a joint research program for DARPA/TTO, and developed a prototype microsensor node called AWAIRS 1 (Adaptive, Wireless Arrays for Interactive RSTA in SUO) [20]. This device consists of modular architecture, in that the modular boards can be freely stacked up on top of the existing system to support additional sensor interfaces. The system features Intel SA1100 microprocessor based on 32-bit ARM RISC architecture, 128KB of SRAM and 1MB of Flash storage, >100m of wireless connectivity at 100Kbps, and 2kHz bandwidth of audio interface. Two 40-pin mini-connectors are used as a system bus, connecting all module boards. The system consumes 1W and the size of a node is  $7 \times 6.7 \times 8.0$  cm<sup>3</sup>. In 2000,  $\mu$ AMPS mote was developed by MIT [21] for the acoustic sensing applications, as shown in Figure 1.1. The work improves the power consumption through the power-aware operation methodology



Figure 1.2: (a) Audio sensor mote for acoustic scenes monitoring (2009) (b) TinyEARS (2013).

such as dynamic voltage scaling, energy-quality trade-off, and fine-grained control over power states of radio module depending on the transmission range.

Afterward, lots of miniaturized wireless sensor platform have been surged for both academic and commercial purposes, such as BTnode (2001) [22], Medusa MK2 (2002) [23], IMOTE (2003) [24], MICA/MICA2/MICA2 (2004) [25], Telos/TelosB (2005) [26], Waspmote (2008) [27], LOTUS (2011) [28], and .NOW eMote (2012) [29]. Although these motes have been improved in terms of their size, performance and power consumption as the Moore's Law continued, their architectural solution remain same as AWAIRS 1, using pluggable modules to create an acoustic sensing interface. None of these works have exploited the opportunity of benefits from built-in integration and optimization for the audio sensing and processing tasks. Instead, they provide suitable ports to allow a variety of sensors to be attached for more versatility.

There yet exists several works to devise miniaturized wireless sensor node dedicated to audio applications. In [30, 31], the authors proposed a sensor mote based on the commercial MICAz [32] platform. The mote shown in Figure 1.2a was developed for voice activity detection (VAD), gender classification, and acoustic feature extraction for further processing in base-station. To process audio signal more ef-

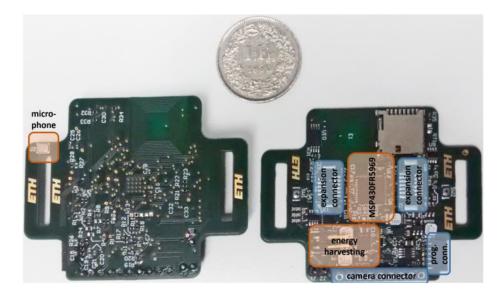


Figure 1.3: InfiniTime (2016).

ficiently, they attached digital signal processing (DSP) module, which includes TI TMS320C6713 DSP chip operating at 225MHz, to the mote. Although it provides profound capability to process various audio tasks in real-time, it draws >500mA of current, and its size is 21 cm  $\times$  11.5 cm, limiting its practical usage. A custom sensor node with self-designed analog front end (AFE) interfacing microphone to fine-grained control over parameters and power consumption was proposed in [33] for acoustic event detection. This system consumes 200mA at active mode, but the  $1.5\mu$ A of sleep mode current and highly duty-cycled operation enable the power supply from a super capacitor and solar harvesting, achieving unattended operation of a sensor node. However, since this node system aimed for infrequent event detection whose sampling rate is similar to that of scalar sensors, its operation under the harvesting is not scalable to general-purpose real-time audio streaming applications, considering its active power level. Moreover, the system size of a few hundreds of  $cm^2$  reduces its efficacy of broad deployment. The authors of [34] developed a wireless audio sensor mote based on IMOTE2 [35] as shown in Figure 1.2b for household appliances power monitoring network based on acoustic signature. The IMOTE2 has a built-in DSP coprocessor with wireless MMX instruction set enabling low power signal processing acceleration. By the benefit from it, the system efficiently implements audio signal processing pipeline such as FFT, MFCC, and feature extraction at 30mA of active current consumption within  $3.6 \times 4.8 \times 1.5$  cm<sup>3</sup> of volume, realizing a centimeter-scale audio sensing. The power consumption of an audio sensor node is further improved in [36] by the holistically customized system design, avoiding the use of any existing platforms. This flexibility of system design allows extreme power management to keep the quiescent and operative energy consumption low. The realtime audio acquisition consumes only 1.2mA, and the system size is approximately 4  $\times$  6 cm<sup>2</sup> as shown in Figure 1.3.

Although a lot of efforts have pursued to reduce the form factor of audio sensor node, they still remain at centimeter-scales. In addition, mA-level of power consumption prevents to further shrink the size of system due to the constraint from battery size. Thus, this dissertation focuses on the realizing of millimeter-scale audio sensor node to enable pervasive audio computing.

### 1.2 Challenges of Achieving Millimeter-Scale Audio Sensing

For many applications of wireless audio sensor network, frequent battery recharging or replacement is unlikely feasible, especially in large scale deployments with thousands of densely distributed nodes, or for nodes placed in where hard to access. Thus, long enough life time under the battery and/or autonomous operation by energy scavengers are indispensable. Meanwhile, the advance in battery technology is lagging [37], as the amount of energy stored in the micro battery is decreased proportionally to its size. This battery size limitation imposes critical constraint on the miniaturization of audio sensor node. Therefore, the energy efficiency and low power operation of every components in the system are critical to guarantee the unattended operation of a sensor node for a long time, and as a corollary, to achieve ultra-small form factor.

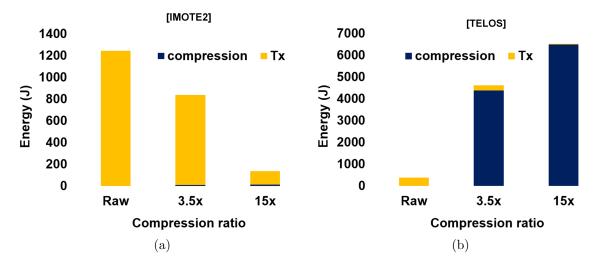


Figure 1.4: Energy consumption vs. compression ratio on (a) IMOTE2 platform and (b) TELOS platform.

Typically, wireless transmission of sensed data over the network dissipates the most of power in a sensor system [38]. Consequently, in addition to the low power wireless protocols and transceivers, the data compression also plays as a key enabler for miniaturized audio sensor node, by reduction of the required network bandwidth. Furthermore, the compression saves more space of system storage, allowing longer audio logging and more complex on-edge post processing. For example, in [35], the authors implemented audio compression algorithm based on linear prediction to prove the benefit of on-sensor compression over the raw data transmission. Depending on the compression ratio, the compression reduced overall energy consumption by  $9 \times$  at the most when  $15 \times$  compression is applied as shown in Figure 1.4a. For this purpose, many researchers have attempted to implement built-in compression on the audio sensor node, such as  $4 \times$  compression by G.726 at 26mA [39],  $16 \times$  compression by Speex at 260mA [40], and  $1.6 \times$  compression by linear prediction at  $\sim 3$ mA [38]. However, all these prior works used off-the-shelf algorithms based on software implementation, and thus the compression itself easily consumes > mA of current even with the very simple algorithm, which provides minimal compression ratio. As shown in Figure 1.4b, even the same compression algorithm and implementation on different hardware platform

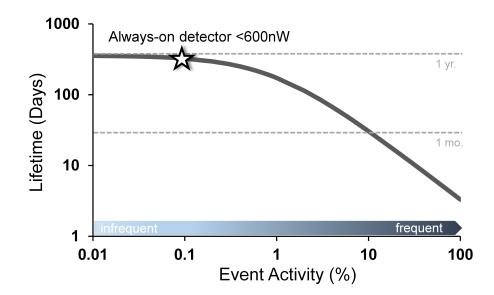


Figure 1.5: Sensor system lifetime vs. event activity: When the event of interest occurs infrequently, the always-on detector would determine the lifetime.

give worse results due to the hardware inefficiency. Therefore, both more efficient compression algorithm and its hardware acceleration should be devised to make the energy and power consumption below the budget of millimeter-scale batteries and harvesters.

Another important challenge is to minimize standby power of overall system since a sensor node stays at standby or sleep mode a vast majority of time. Such low energy resource systems rely heavily on the duty-cycled operation to ensure long lifetime especially for audio sensing applications where the event of interest occurs infrequently. As the standby power dominates over time, the leakage current of each system component should be minimized. In addition, deliberated system wakeup method should be considered to use limited resources more efficiently. Conventionally, the audio sensor motes include low power wakeup timer to wake system up with a predefined period [33, 34]. However, this periodic wakeup may miss the events or waste resources at out-of-events. Instead of predetermined wakeup, the host can send a query to a node by network, and then the low power wireless wakeup receiver could trig the system up to start the tasks [38]. Although this method delivers high flexibility to the host or user when to wake the system up, it charges a burden of careful managing strategy of whole network to the host, resulting in high system complexity and costs. Moreover, this method also fails to avoid the event hit and miss problem without prior knowledge about the event rate and instances. Therefore, the development of more efficient yet low power wakeup method for audio applications is highly required. The wakeup method should intelligently detects the event of interest so that the sensor system knows by itself the event rate and instances. However, at the same time, the wakeup method should also consume minimal power since the always-on wakeup detector would dominate overall average system power when the event of interest occurs infrequently as shown in Figure 1.5.

Finally, since the wireless audio sensor node consists of many COTS components such as microphone, quartz crystal, antenna, and solar-cells for autonomous operation, achieving millimeter-scale form factor is extremely challenging. Thus, the physical integration process of a system must be carefully devised to reduce the size dramatically.

#### **1.3** Dissertation Overview

This dissertation seeks to develop a millimeter-scale wireless audio sensor node enabled by various low-power integrated circuit architectures and designs, addressing the aforementioned challenges.

In Chapter II, the design of low-power audio processing chip is studied. Along with analog front-end circuits to interface with a MEMS microphone, a novel audio compression algorithm is proposed and the hardware architecture for the acceleration of it is discussed, resulting in  $4-32\times$  real-time audio compression while consuming  $1.5\mu$ W.

Chapter III introduces an ultra-low standby power neural (NN) network processor. As the ML and AI show rapid progress, there is a surging need for built-in processing capability of ML workloads on sensor node. The proposed on-sensor NN processor meets the need while consumes only 440pW of standby power, enabling long lifetime of duty-cycled, miniaturized sensor systems. The proposed processor is integrated in acoustic sensor system and demonstrates >90% of detection accuracy for multiple acoustic objects in real-time.

Chapter VI discusses acoustic wakeup methods for the audio sensor nodes. A lowpower voice activity detection (VAD) technique is proposed to make a human speech activity as a target wakeup event. The design of VAD chip is detailed and validated in 180nm CMOS technology. Moreover, a wakeup method based on acoustic signature is also studied in this Chapter. The proposed VAD consumes 140nW of power with >80% of accuracy at 50 dBA SPL of sound, and the acoustic signature detection consumes only 66nW.

Finally, two generations of millimeter-scale wireless audio sensor nodes are demonstrated in Chapter V, with the system design and integration strategy.

### CHAPTER II

# A Microwatt Power Audio Processing IC and 8Mb Streaming Flash Memory

#### 2.1 Introduction

Realizing a millimeter-scale audio processing platform enables a number of new IoT applications such as distributed audio recording, event logging, and security monitoring. While several efforts [39, 40] have sought to miniaturize audio sensors, their centimeter-scale volume and >20mW power severely limit use as an unobtrusive, selfpowered sensing node. The key challenge to reduce the form factor of audio sensor node is low power operation due to the small battery size as shown in Figure 2.1. For example, the audio mote consuming 330mW [39] can only last 0.7s with millimeterscale thin-film Li battery ( $16\mu$ Ah, 4V). In addition, small physical size severely limits the capacity of data storage. As shown in Figure 2.1, a 1×0.85 mm<sup>2</sup> NOR Flash has less than 2Mb density [41]. Consequently, power efficient data compression is a key enabler in that it not only mitigates the size requirement of on-site storage, but also reduces wireless transmission energy. However, 22mA of current consumption in [40] implies that compression itself requires high computation burden. One another enabler for small form factor of audio sensor node is a power efficient compact non-volatile storage to save audio footage. Since a sensor node is typically highly

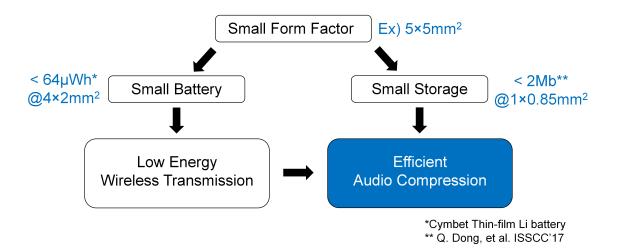


Figure 2.1: Miniaturized audio sensor challenges.

duty cylced, it spends only a fraction of their time in active mode. Therefore, embedded Flash memory is a good candidate to store measured audio data, allowing the other system components to be power-gated in standby mode. This chapter demonstrates an audio processing IC that consumes only  $4.7\mu$ W for signal acquisition and compression. The proposed compression engine operates in real-time at  $1.44\mu$ W. In addition, this chapter also introduces an 8Mb custom NOR Flash for continuous audio streaming and retention.

#### 2.2 Audio Processing IC

Figure 2.2 shows the architecture of the proposed audio processing IC that integrates AFE, ADC and compression engine. First, we use capacitive MEMS microphone, and it requires 10V bias voltage for the optimal sensitivity [42]. The charge pump generates bias voltage directly from battery voltage, VBAT (3.6 - 4.2V). The AFE and ADC operate at 1.4V and 0.9V respectively, regulated by on-chip LDOs from the battery to decouple them from the noisy digital supply. To optimize the power consumption, we employ multi-voltage, multi-threshold design strategy for the digital circuits. The compression engine operates at 0.6V with standard threshold transistors

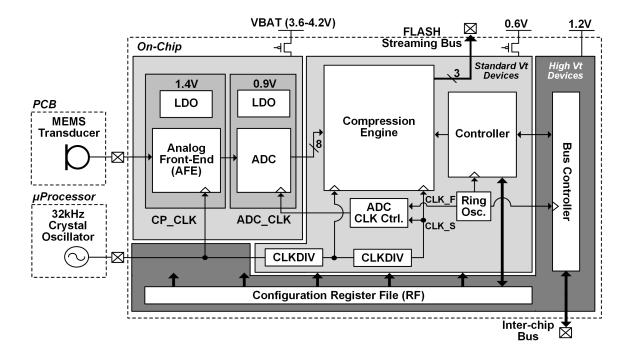


Figure 2.2: Overall architecture of audio processing IC.

to meet the real-time throughput constraint. These logic blocks are power-gated in sleep mode. The 1.2V bus controller and configuration register file are always-on and thus use thick oxide I/O devices to reduce leakage current.

Table 2.1: Measurement Sum	nmary of A	nalog Front-End
This Work		
Supply	$1.4\mathrm{V}$	
Gain	20-48dB	
Bandwidth	4kHz	
Input referred noise (IRN)	$13.2\mu V_{rms}$	$_{s}$ (A-weighted)
	LNA	$1.1 \mu A$
Current	VGA	$0.8 \mu A$
	Total	$1.9 \mu A$
NEF		11.1

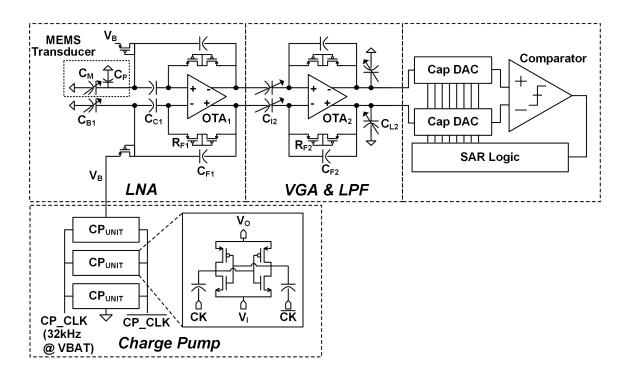


Figure 2.3: Analog front-end (AFE) and ADC circuits.

#### 2.2.1 Analog Front-End Design

The analog front-end (AFE) shown in Figure 2.3 consists of low noise amplifier (LNA), variable gain amplifier (VGA) and charge pump. The charge pump biases the MEMS transducer at 10V. It is based on 3-stage voltage doubler circuits and generates bias voltage with 32kHz of clock. Note that to realize differential structure, dummy capacitors and resistors (pseudo) are added on the other side of microphone, matching the input impedance. The LNA gain (29dB) is set by the MEMS capacitance over the feedback capacitance ( $C_M/C_{F1}$ ), and the gain and bandwidth of VGA are tuned by capacitor arrays  $C_{I2}$  and  $C_{L2}$ , respectively.  $R_{F1,2}$  sets input common-mode voltage and removes offset. To maximize noise efficiency, OTA<sub>1</sub> and OTA<sub>2</sub> use inverter-based cascode amplifier and their input-pair transistors operate in subthreshold region. The bandwidth of this analog gain stages was set to 4kHz. Figure 2.4 shows measured input referred noise (IRN) spectrum of the AFE. The LNA and VGA consume 1.54 $\mu$ W and 1.11 $\mu$ W, respectively, to achieve 20.1 $\mu$ V<sub>rms</sub> (non-weighted) and 13.2 $\mu$ V<sub>rms</sub> (A-

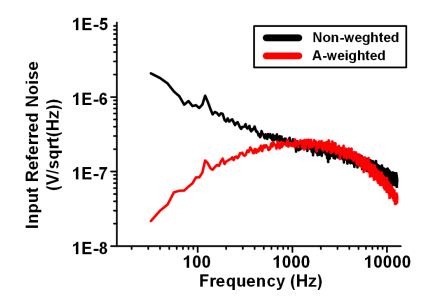


Figure 2.4: Measured input referred noise spectrum of AFE.

weighted) IRN with 4kHz of bandwidth. The measurement results of the AFE is summarized in Table 2.1. After gain stages, the 8-bit synchronous SAR ADC (Figure 2.3) digitizes the incoming amplified and filtered audio signal. The ADC operates on two separate clocks: an 8kHz clock (CLK\_S) for sample and hold, and a 150kHz clock (CLK\_F) for the internal SAR control logic. Since the audio signal is processed in the frequency domain at compression engine, the frequency stability and phase noise of CLK\_S can directly affect the audio compression quality while CLK\_F has significantly relaxed constraints. The 32kHz external crystal clock is divided to 8kHz (CLK\_S) and merged with internal 150kHz clock (CLK\_F) obtained from a powerefficient ring oscillator at 0.6V by the ADC clock controller as shown in Figure 2.5. With this controller scheme, the need of high frequency crystal oscillator for the synchronous SAR ADC clocking can be vanished. The measured SNDR of ADC is 46.1dB which corresponds to 7.4-bit ENOB, and the FoM is 100fJ/Conv as shown in Figure 2.6. The ADC consumes only 135nW of power. Table 2.2 summarizes the measurement results of ADC.

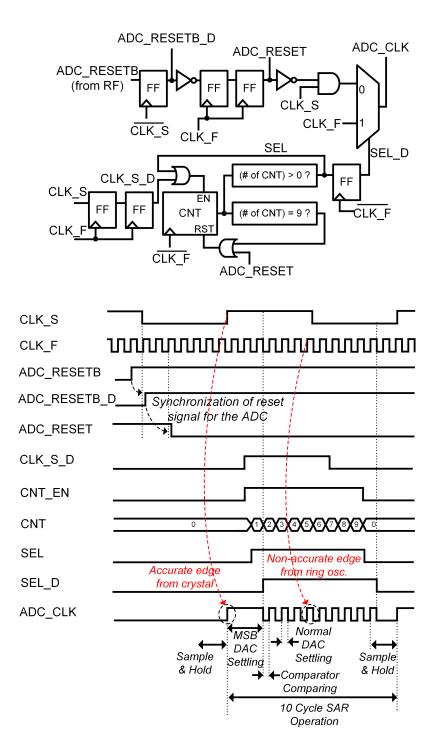


Figure 2.5: Synchronous SAR ADC clock controller circuits and its operational timing diagram.

	This Work				
Supply	$0.9\mathrm{V}$				
$\mathbf{Fs}$	8kHz				
INL	+0.26/-0.34 LSB				
DNL	+0.27/-0.17 LSB				
SNDR	46.1dB				
ENOB	7.4bits				
Power	$135 \mathrm{nW}$				
FoM	$100 \mathrm{fJ/Conv}$				

Table 2.2: Measurement Summary of ADC

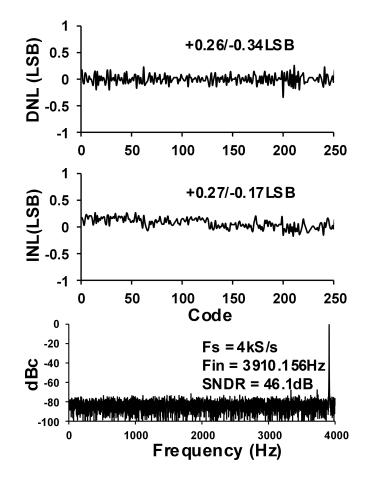


Figure 2.6: Measured DNL, INL, and frequency spectrum of ADC.

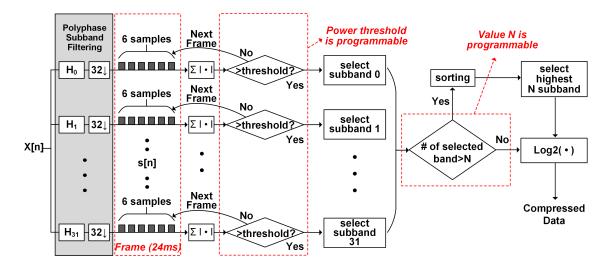


Figure 2.7: Proposed compression algorithm.

#### 2.2.2 Compression Algorithm

Compression of the audio stream is critical to minimize storage size, access energy, and wireless transmission energy. However, real-time audio compression is a computationally intensive task. Thus compression power itself should be minimized for the proposed ultra-small audio sensor node.

The proposed power efficient compression algorithm is shown in Figure 2.7. The incoming samples are first converted to the frequency domain using polyphase subband filtering (PSF). This filtering technique critically samples the incoming signal by modulating and decimation, meeting Nyquist criteria for each center frequency of band-pass filter in filter bank. By nature, this filter technique reduces the amount of data by eliminating overly sampled portions. However, due to the non-ideal band-pass filter characteristic, signal aliasing always occurs, and therefore it results in a lossy compression. In contrast to the block transform such as FFT, this filter bank overlaps in time domain and avoids block edge artifacts. Once the signal is filtered by equally spaced 32 filter bank, the power of each subband is accumulated during 1 frame which consists of 6 samples. And then, each accumulated power is compared with a programmable power threshold. The subbands having lower power than a thresh-

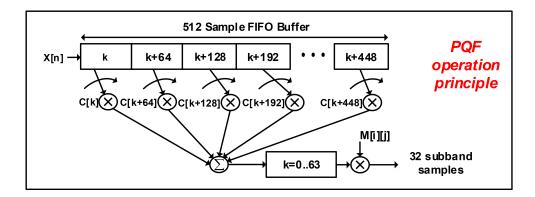


Figure 2.8: Polyphase Quadrature Filtering (PQF) process.

old are eliminated, and we only select subbands whose power exceed the threshold. Then, the number of selected subbands is compared with a pre-determined (but, programmable) number N. If the number of remaining subbands exceeds the number N, the highest power of N subbands are selected by sorting. This processes guarantees a constant worst-case compression rate but, in general, the final compressed bit rate can vary depending on the frequency domain sparsity of the signal. Finally, logarithmic-compressed quantization is applied to further reduce data rate.

In this algorithm, the complexity of the PSF dominates others. To have linear phase shift to prevent any phase distortion for audio quality, 512-tap FIR filter is used to realize band-pass filters. The filtered signal can be represented as follows:

$$s_t[i] = \sum_{n=0}^{511} x[t-n] \times H_i[n]$$
(2.1)

where  $H_i$  is an impulse response of the *i*-th band-pass filter, x[t] is an audio sample at time *t*, and  $s_t[i]$  is the filter output sample for subband *i* at time *t*, where *t* is an integer multiple of 32 audio sample intervals. In the PSF, the band-pass filter can be regarded as a modulated version of a prototype low-pass filter as follows:

$$H_i[n] = h[n] \times \cos[\frac{(2i+1)(n-16)\pi}{64}]$$
(2.2)

where h[n] is a prototype low-pass filter. This plain implementation of PSF requires 1023 OP/sample of complexity. To reduce power, we apply a mathematicallyequivalent but more computationally efficient polyphase quadrature filtering (PQF) [43]. Since the sinusoid in Equation 2.2 contains an odd number of half cycles in 64 coefficients, blocks of 64 products of the multiplications are accumulated with the sign of alternate blocks negated. These 64 points are then multiplied by 32 sinusoids to generate 32 output samples. This process is depicted in Figure 2.8. From the explained PQF process, we can derive the following equation for the filter bank outputs:

$$s_t[i] = \sum_{k=0}^{63} \sum_{j=0}^7 M[i][k] \times (C[k+64j] \times x[k+64j])$$
(2.3)

where

$$C[n] = \begin{cases} -h[n] & n/64 = odd \\ h[n] & \text{otherwise} \end{cases}$$
(2.4)

$$M[i][k] = \cos\left[\frac{(2i+1)(k-16)\pi}{64}\right]$$
(2.5)

This implementation requires 157 OP/sample, which is 84% reduction in the complexity. Then, this filtering is further optimized by inverse discrete cosine transform (IDCT) conversion [44]. Equation 2.3 can be re-written as follows:

$$s_t[i] = \sum_{k=0}^{63} y[k] \times \cos\left[\frac{(2i+1)(k-16)\pi}{64}\right]$$
(2.6)

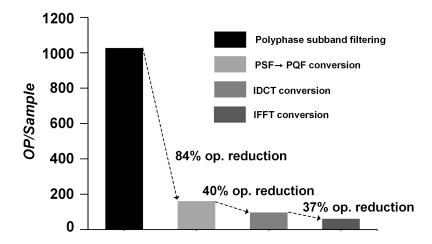


Figure 2.9: Compelxity reduction from the algorithm optimization.

By re-arranging the y[k] sequence, the above equation is re-formulated as follows:

$$s_t[i] = \sum_{k=0}^{63} y'[k] \times \cos\left[\frac{(2i+1)k\pi}{64}\right]$$
(2.7)

where

$$y'[k] = \begin{cases} y[16] & k = 0\\ y[k+16] + y[16-k] & k = 1, 2, ..., 16\\ y[k+16] - y[80-k] & k = 17, 18, ..., 31 \end{cases}$$
(2.8)

Equation 2.7 is well-known IDCT fomula. This optimization reduces the complexity to 93 OP/sample. Finally, IDCT computation is performed using a mathematically equivalent inverse FFT (IFFT) operation as follows:

let 
$$X[k] = y'[k] \times e^{jk\pi/2N}$$
  $k = 0, 1, 2, ..., 31$  (2.9)

$$x[m] = Re[FFT^{-1}(X[k])] \quad m = 0, 1, 2, ..., 31$$
(2.10)

Table 2.3: Compression Algorithm Comparison				
	MPEG Layer III	CELP	$ADPCM^1$	This Work
Raw Data $Rate^2$		64kbps		
Compressed Data Rate	32kbps	$9.6 \mathrm{kbps}$	$16 \mathrm{kbps}$	$4.07 \mathrm{kbps}$
Compression $\operatorname{Ratio}^3$	2	6.7	4	15.7
Latency Variable Bit Rate	>100ms Yes	30ms Yes	$\stackrel{<2ms}{No}$	32ms Yes(2-16kbps)
Complexity	3000  op/sample	45000  op/sample	4  op/sample	61  op/sample
$ODG^4$	0.2	-3.873	-3.885	-3.879

Table 2.3: Compression Algorithm Comparison

<sup>1</sup>Linear ADPCM with 1'st order prediction

<sup>2</sup>Male voice in English

<sup>3</sup>Raw data rate/compressed data rat

<sup>4</sup>Objective Difference Grade; -4(very annoying) - 0(imperceptible)

then 
$$s_t[i] = \begin{cases} s_t[2i] = x[i] & i = 0, 1, 2, ..., 15 \\ s_t[2i+1] = x[31-i] & i = 0, 2, ..., 15 \end{cases}$$
 (2.11)

By the efficiency of FFT computation, the complexity is reduced to 57 OP/sample. Overall, the complexity is reduced by 94% in total from aforementioned algorithm optimizations as shown in Figure 2.9. A comparison between the proposed and other off-the-shelf algorithms is shown in Table 2.3. Maintaining similar sound quality, the proposed algorithm has  $1000 \times$  lower complexity than CELP and  $3.9 \times$  better compression than ADPCM.

#### 2.2.3 Compression Engine Architecture

The fixed-point pipelined architecture is proposed to implement the algorithm efficiently as shown in Figure 2.10 and Figure 2.11. Multiple word widths with different fixed-point positions are applied to balance between the power consumption and truncation error.

The proposed polyphase quadrature filter implementation is shown in Figure 2.10. To realize polyphase quadrature filter, previous 512 samples must be latched to be processed while new samples are continuously streamed in at sampling rate, which

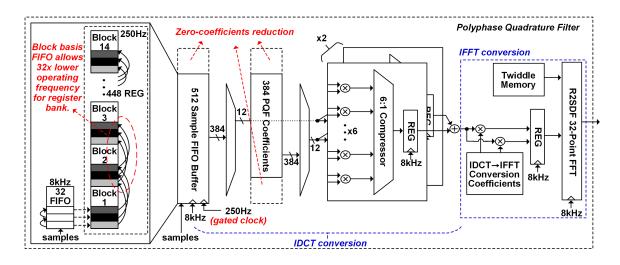


Figure 2.10: Polyphase quadrature filter architecture.

requires 512-entry FIFO shifted at 8kHz consuming 62% of total compression power. To address this predominance, new samples are streamed into 32-entry FIFO instead, and shift operation of whole samples is performed block-by-block only when 32-entry FIFO becomes full. Consequently, 512-entry FIFO is clock gated until the next block shift, resulting in  $32 \times$  power reduction. In addition, we observe that 25% of filter coefficients are zero, allowing us to inactivate unused samples and to eliminate corresponding registers and datapath. The 512 samples are re-arranged and R2SDF based 32-point FFT is performed to realize the optimized subband filtering.

Filtered data are then stored in a sub-frame buffer as shown in Figure 2.11a. Since compression is performed in a frame (i.e., 6 sub-frames) basis, the two additional buffers are added (i.e., total 8 sub-frame buffers) as ping-pong buffers to store next frame data while processing the current frame. The sub-frame buffers and power accumulators are clock gated with frame based operation, and this clock gating shows 34% of total engine power reduction (simulated). Moreover, along with clock gating, the data gating is used to avoid unnecessary data switching on shared data bus of sub-frame buffers. This data gating further reduces the power consumption by 25% (simulated).

For frequency-rich signals, sorting operation is necessary. The proposed sorting

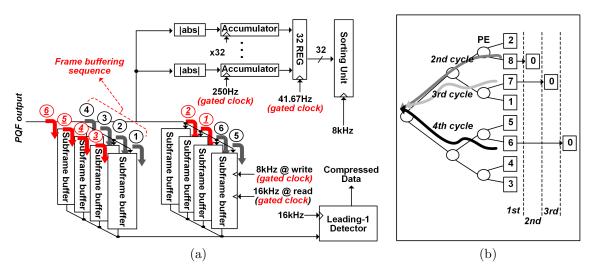


Figure 2.11: Compression engine architecture.

unit uses a tree structure as shown in Figure 2.11b), where all processing elements (PEs) compare and forward their inputs in the 1st cycle to obtain the top result. Hence, every PEs are active in the 1st cycle. Then, in each subsequent cycle the winning PE zeros its value and only its related path is updated to produce the next highest values. So for the remaining cycles until finding the top N out of 32, only a fraction of PEs are in active. Compared with a conventional parallel sorter, such as bitonic sorting, this implementation shows 42% less dynamic energy for sorting top 16 out of 32. The complexity of the proposed sorting when finding the top N/2 out of N is described as follows:

$$(N-1) + \frac{N}{2} \times \log_2 N \tag{2.12}$$

After pruning, selected subband samples are log-domain quantized with leading-one detector, implemented with round approximation.

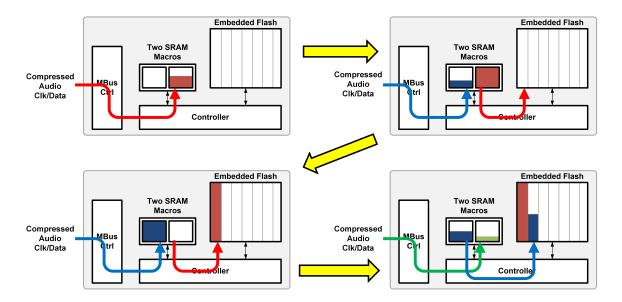


Figure 2.12: Operation principle of ping-pong streaming Flash.

#### 2.3 8Mb Embedded Streaming NOR Flash

One key requirement to realize a miniature audio sensor node is embedded nonvolatile memory for compact and retentive storage of sensed audio data. Non-volatile storage retains the data even during power source outage. Also, it allows near-zero retentive power during the standby mode, and thus lengthens the lifetime of a sensor node. In general, however, typical NOR Flash memory consumes mW-level of power for program and erase, preventing its usage as an embedded storage for sensor applications. To address this issue, [41] proposed ultra-low power 1Mb NOR Flash design. They increased the efficiency of high voltage generation, which is often a power bottleneck during program, by using a combined Dickson and Cockcroft-Walton charge pump with MIM capacitors. In addition, they employed separate power gating for each bank so that the peripheral circuits of the banks that are not used can be powergated. With other low power circuit techniques, their Flash design consumes  $\mu$ W-level power consumption for program, read, and erase operations.

In this work, we design a custom 8Mb NOR Flash chip following all the strategies in [41]. As the capacity and size of Flash macro increase, one distinctive feature is proposed and designed in this work, since the instantaneous power for program exceeds the allowed budget of sensor system. Although the compressed audio data rate in Section 2.2 is <7kbps in average (measured results are shown in Section 2.4), the streaming format requires 32kbps burst data rate since the compression algorithm operates over multiple sub-frames. To meet this program speed, the 8Mb Flash chip should consume more than twice of power compared with 1Mb macro in [41]. To solve this issue, we design a streaming Flash with ping-pong SRAM buffers as shown in Figure 2.12. Since the SRAM write power is much less than the Flash, we buffer the incoming stream into the SRAM first with higher write speed. Two SRAM macros act as ping-pong buffers: one macro filled with audio streaming; the other being transferred to Flash. In this way, we can keep the Flash program cycle time longer than requirement to reduce instantaneous power consumption. The Flash controller automatically handles seamless ping-pong streaming.

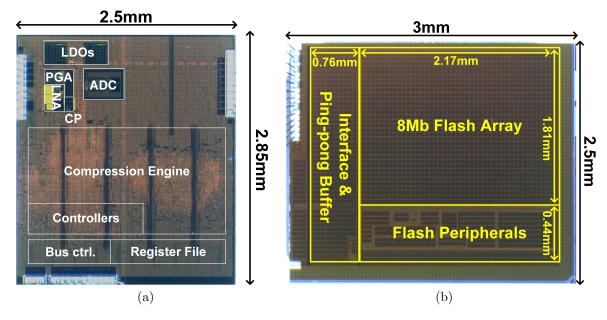


Figure 2.13: Chip die photo of (a) audio processing chip and (b) 8Mb NOR Flash chip.

## 2.4 Measurement Results

The proposed audio processing chip is fabricated in 180nm CMOS with an active area of 7.125mm<sup>2</sup>, and the die photo of it is shown in 2.13a. Measured A-weighted input referred noise of amplifiers is  $13.2\mu V_{rms}$  (Figure 2.4), translates to 61dBA of SNR at 94dBSPL (1kHz) input sound. We also performed acoustic testing as shown in Figure 2.14a. The audio chip is integrated with a MEMS microphone on the chipon-board (COB) setting, and placed inside anechoic chamber to measure acoustic performance accurately without ambient sound noise. Playing a tone sound using a speaker and measure its sound pressure level (SPL) by a reference microphone. In this acoustic testing, the analog front-end consists of MEMS microphone, charge pump, LNA and PGA. The overall analog chain achieves  $32\mu V_{rms}$  A-weighted input referred noise while consuming  $3.15\mu$ W of power. The SNR and sensitivity are 54.6dBA and  $-14\sim13.6$ dBV, respectively, with 94dBA SPL input sound at 1kHz. Figure 2.14b shows the power spectral density (PSD) of this analog front-end.

Measured compression rate depends on the signal sparsity. Figure 2.15 shows the

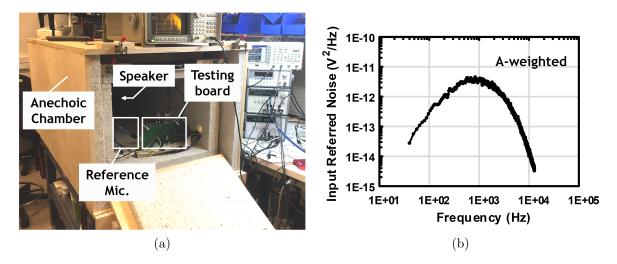


Figure 2.14: (a) Acoustic measurement setup (b) power spectral density (PSD) of AFE.

Table 2.4: Measurement Results of Compression Ratio							
Compressed Data Rate Compression Rat							
Tone	$2.69 \mathrm{kbps}$	$23.8 \times$					
Slow speech	$3.56 \mathrm{kbps}$	$17.9 \times$					
Normal speech	$4.37 \mathrm{kbps}$	$14.6 \times$					
Music	$6.71 \mathrm{kbps}$	9.5  imes					

time domain plots of compressed and original audio clips. Figure 2.15a represents normal speed of speech case, and Figure 2.15b shows slow speech case for same sentence. We can observe that the compressed signal samples are highly suppressed in quiet region since they are below the power threshold. For a normal speech, measured average compression ratio is  $14.6\times$ . On the other hand the compression ratio is  $18\times$  for a slow speech, as expected. Table 2.4 summarizes the compression ratio for various sound types, and it varies automatically depending on the signal sparsity. For the normal human speech, the audio processing chip provides  $\sim 15\times$  compression, enabling >30 mins of recording with 8Mb custom Flash.

Compression rate and quality trade-off can be tuned by the number N and the power threshold. In Figure 2.16, the x-axis is the power threshold, the value below which subbands are pruned for compression. A higher threshold means more aggres-

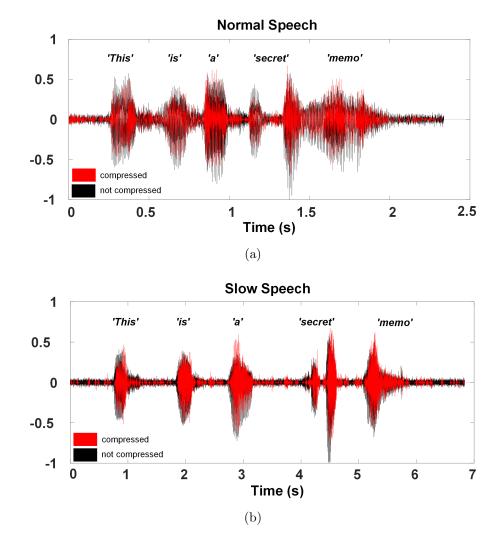


Figure 2.15: Time domain signals for compressed and origial audio clip at (a) normal speed speech and (b) slow speed speech.

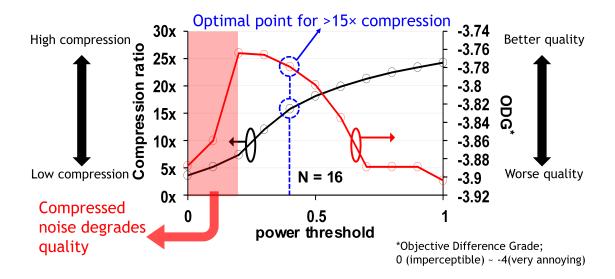


Figure 2.16: Measured compression ratio vs. sound quality trade-off.

sive compression. The y-axis on the left is the compression ratio, and the y-axis on the right is sound quality: A higher value indicates the better quality. The number N is the maximum number of remaining subbands after pruning, and we are using N=16 case for this measurement. As the power threshold is decreased, audio quality improves, but compression rate also decreases as expected. However, when the power threshold goes too low, noise gets to be included in compression, and degrades the audio quality.

The audio processing IC consumes  $4.73\mu$ W of power, dictated by continuous run for ~14 hours under  $64\mu$ Wh millimeter-scale battery. Figure 2.17a shows the power breakdown of the audio processing IC. The compression engine dissipates  $1.44\mu$ W of power, and its power breakdown is shown in Figure 2.17b. As shown in the Figure, polyphase subband filtering (pre-processing + FFT) consumes 74% of total compression engine power even after all of optimizations from both algorithm and architecture due to the high amount of data to process and the required throughput. Table 2.5 summarizes the proposed audio processing IC.

The proposed custom 8Mb embedded Flash is fabricated in 90nm ESF3 NOR Flash technology. Due to its increased capacity, program power becomes double

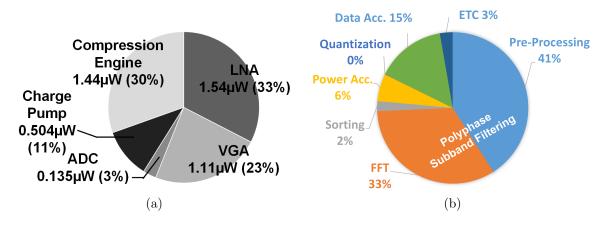


Figure 2.17: Measured power breakdown of (a) audio IC and (b) compression engine.

Table 2.5: Summary of Audio Processing IC					
	This Work				
Technology	180nm				
Die area	$2.5 \times 2.85 \mathrm{mm}^2$				
Supply voltage	0.6/1.2/3.6V				
Amplifiers gain	20-48dB				
AFE bandwidth	4kHz				
AFE input referred noise	$32\mu V_{rms}$ (A-weighted)				
AFE SNR	54.6 dBA @ 1Pa				
Clock frequency	$8/16/150 \mathrm{kHz}$				
Compression ratio	$4-32 \times$ (auto-variable)				
Power	$4.73 \mu W$				

Table 2.6: Comparison of Embedded Flash ICs							
	This Work		[41]	[45]	[46]		
Technology	90nm ESF3		90nm ESF3	90nm MONOS	$130 \mathrm{nm}$ SONOS		
Capacity	8Mb		1Mb	1Mb	$1024 \times 260 \mathrm{b}$		
Area $(mm^2)$	7.5		0.73	2.26	0.85		
Prog. throughput	$730 \mathrm{kbps}$	$7.2 \mathrm{kbps}$	$800 \mathrm{kbps}$	$341 \mathrm{kbps}$	$1 \mathrm{Mbps}$		
Prog. power	$80\mu W$	$38 \mu W$	$39 \mu W$	$323\mu W$	$125 \mu W$		
Prog. energy	$110 \mathrm{pJ/bit}$ $656 \mathrm{pJ/bit}$		$49 \mathrm{pJ/bit}$	$946 \mathrm{pJ/bit}$	$122 \mathrm{pJ/bit}$		
Read power	$28 \mu W$		$25 \mu W$	N/A	$176 \mu W$		
Read energy	3.5p.	J/bit	$3.1 \mathrm{pJ/bit}$	N/A	$1.2 \mathrm{pJ/bit}$		
Erase power	$13 \mu W$		$15 \mu W$	N/A	N/A		
Erase energy	$3.3 \mathrm{pJ/bit}$		$9.4 \mathrm{pJ/bit}$	$1.07 \mathrm{nJ/bit}$	$29 \mathrm{pJ/bit}$		
Standby power	7.8	$\mu \mathrm{W}$	$5.4 \mu W$	N/A	N/A		

than [41] when it uses similar program throughput. However, thanks to the pingpong streaming technique, we can tune down the program speed to 7.2kbps, reducing the program power by 2.1×. We achieve  $38\mu$ W of program power which is the lowest, compared with prior works for custom embedded Flash as shown in Table 2.6. Moreover, 8Mb of large capacity enables >30 mins of continuous audio streaming when combined with the proposed compression engine. Read and erase consumes  $28\mu$ W and  $13\mu$ W of power, respectively, also meeting the peak power budget of millimeter-scale energy sources.

Table 2.6: Comparison of Embedded Flash ICs

## CHAPTER III

# A Picowatt Standby Power Neural Network Processor With Custom ISA and 7T SRAM for Sensor Applications

## 3.1 Introduction

Wireless sensor node enables remote retrieving of video and audio streams, images, and scalar sensor data such as temperature, pressure or humidity. To be benefit from massively distributed sensor nodes, the size reduction of each sensor device is paramount. Minuscule size makes the sensors as easy-to-deploy and unobtrusive as possible, minimizing the disturbance to human activity. Continued advance of integrated circuits and technologies have enabled millimeter-scale sensing nodes [1,3, 7,47], paving the way to ubiquitous wireless sensor applications.

As machine learning (ML) techniques such as neural network (NN) have been rapidly proliferated, having built-in NN processing capability on sensor becomes a highly desirable feature to implement efficient large-scale sensor network, alleviating bandwidth, latency, security and communication energy limitations [48, 49]. While the NN workloads require intensive computation and large memory footprint, the sensor node design is heavily constrained by its small form factor, energy, and storage. The software implementation of NN processing on the general-purpose microprocessor typically equipped with sensor nodes lacks energy and memory efficiency, failing to meet the stringent constraints of millimeter-scale node. As a complement, many of prior works have sought to realize dedicated NN accelerators for embedded platforms [50–53]. However, they mainly aimed at convolutional neural networks (CNNs) or large, deep-layered neural networks (DNNs) that can be regard as too excessive capacity for sensing applications [54], and thus their designs rather cause inefficiencies such as low utilization of resources and leakage dominance in the cases where moderately-sized network model is sufficient. Moreover, since prior accelerator designs trade flexibility for efficiency, they can not support not only arbitrary NN topology, but also arbitrary algorithms (e.q., moving average, normalization, sorting, etc.).

In this Chapter, we explore a compact and programmable NN processor to support small-to-medium size models under highly resource-constrained sensor platform. Several prior works optimized for modest NN workloads have been reported in [55–57, 61, 73]. Unlike these works, we consider standby power as the primary design metric since this is of particular importance in a typical wireless sensor node that stays at standby mode a vast majority of time as the event of interest occurs infrequently. As shown in the Figure 3.1a, when the sensor node spends an extended period of time at standby mode, the average power consumption of a sensor node is governed by the processor's standby power. Once the processor is initially programmed using on-chip memory, it is necessary to retain the instructions and data (i.e. NN parameters) during the standby mode to avoid off-chip memory access at every wakeup. The low standby power is also beneficial in always-on sensor, which is the case in Figure 3.1b. Since a typical sensor collects data stream for a long interval to make an input vector to NN (i.e. a frame), the processor computes NN workloads at minimum energy point (MEP) quickly, and then returns to standby mode waiting for next input vector. If the frame length is long enough and the required NN

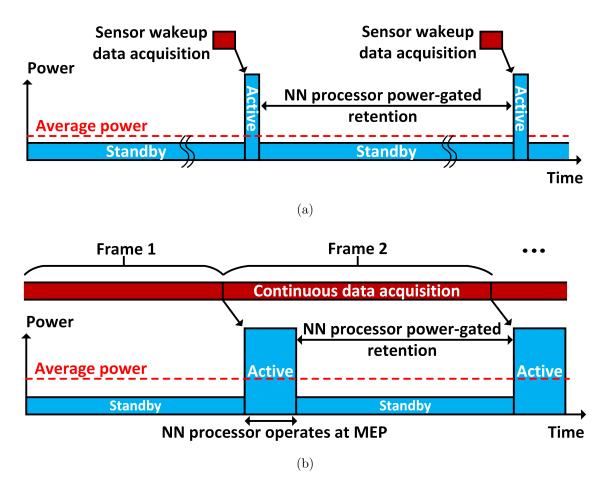


Figure 3.1: The power profile and average power consumption of NN processor in (a) duty-cycled operation and (b) always-on operation.

workload is low-to-moderate, then again the standby mode might contribute more energy [58].

We have developed a programmable NN processor with 440pW of standby power to support small-to-medium NN workloads in the sensor nodes. To reduce the standby power consumption, we exploit a custom instruction set architecture (ISA) for high code density and sufficient flexibility, a lightweight SIMD microarchitecture, and an ultra-low leakage SRAM bitcell. Finally, we have embedded the processor into a single chip integrated sensor system for acoustic object detection to validate the efficacy of the on-sensor NN processor.

The remainder of this Chapter is organized as follows. Section 3.2 introduces the on-sensor NN processor with custom ISA, microarchitecture, and memory cell. Section 3.3 describes the integrated sensor system on chip for acoustic object detection. Section 3.4 discusses the measurement results.

## 3.2 On-Sensor Neural Network Processor

#### 3.2.1 Custom Instruction Set Architecture

The design of a NN processor ISA must consider a few characteristics particularly for modest workloads in highly resource-constrained sensor environment. Since the data retention of on-chip storage dominates overall power consumption at standby mode, the program code size should be compact enough to enable minimal sized memory, and thus minimum data retention power in the processor. At the same time, the ISA must also maintain sufficient flexibility to realize a variety of network model structures/sizes, and additional general-purpose programs as needed. The CISCtype ISA generally provides high code density, resulting in succinct programs to be stored. However, it requires more complex hardware implementations and prevents fine-grained management of hardware resources by programmer, often failing to meet

Instruction	Assembly Format	Description
Load	LD regID,\$dst,len,\$src	load data from memory to register
Store	ST regID,\$src,len,\$dst	store data from register to memory
Vector Add	VADD srcID,len,\$src or #imm	(vector) + (vector/scalar/immediate)
Vector Subtract	VSUB srcID,len,\$src or #imm	(vector) - (vector/scalar/immediate)
Vector Multiply	VMLT srcID,len,\$src or #imm	$(vector) \times (vector/scalar/immediate)$
Scalar Add	SADD srcID,\$src or #imm	(scalar) + (scalar/immediate)
Scalar Subtract	SSUB srcID,\$src or #imm	(scalar) - (scalar/immediate)
Scalar Multiply	SMLT srcID,\$src or #imm	$(scalar) \times (scalar/immediate)$
Matrix-Vector Multiply	MMLT iLen,oLen,shb,shl or shr,\$src,\$dst	$(matrix) \times (vector)$ with shifting
PWL Approximate	PL regID,\$src1,len,\$src2,plshb	non-linear functions on register
Element-wise Compute	EC funcID,len,\$dst	element-wise functions on vector register
Conditional Branch	CBR \$src,#imm,jumpAddr	jump when (reg[\$src]==#imm)
Direct Jump	JMP jumpAddr	direct PC jump to jumpAddr
No Operation	NOP cycles	pipeline stall for cycles of clock cycles

Table 3.1: Instruction Set of On-Sensor NN Processor

stringent power budget. On the other hand, while the RISC-style ISA allows simpler microarchitectures, it consumes more on-chip storage for the same program than CISC approach. Moreover, atomic primitive instructions put more burden on fetch, decode, and rename stages than the computation itself, causing energy inefficiency [59], and this is exacerbated in the case where small footprint of network models are deployed. Therefore, the ISA design should try to keep a balance between both approaches by enabling small number of code lines and low power hardware implementations, simultaneously. Finally, some degree of SIMD data-level parallelism should also be exploited to make the NN computation more efficient.

Taking account of aforementioned characteristics, we define a set of instructions for a programmable NN processor for various sensor applications, and provide the list and assembly format along with their description in Table 3.1. The symbol \$ denotes register and memory addressing, and the symbol # denotes the immediate constant. The instructions consist of fixed 32-bit width: 6 bits for opcode (redundant bits are reserved for instruction extension) and the remaining 26 bits for different fields per instruction. The ISA supports separate scalar and vector register files (RF) which can be shared as either general-purpose registers or NN buffers within the same program, saving the total number of registers. The ISA design also includes on-chip scratchpad memory to store all the data used by NN models. Since NN techniques often require variable-length, contiguous vector or matrix data, the exposure of scratchpad memory to the programmer allows high flexibility for efficient data management. Although the idea of scratchpad memory is similar to [60], their ISA design doesn't support vector RF at the same time and thus, cannot exploit any data reuse opportunities. The vector RF combined with scratchpad memory provides stationary input vector reuse and flexible-width of matrix data access, simultaneously. The ISA is based on a load-store architecture in which the on-chip scratchpad memory access occurs with load (LD) and store (ST) instructions. Since the ISA supports both scalar and vector registers, regID specifies the type of register. The variable-length of data access for vector or matrix can be controlled by len field.

The arithmetic instructions can be used for program control flow, simple algorithm realization, and vector-wise processing. The scalar instructions (SADD, SSUB, SMLT) operate on scalar registers or immediate value in the instructions. The first operand and destination are implicitly fixed to a same register (sreg[\$0]) to reduce hardware complexity and to shorten the length of instructions. Whether to use a value in scalar register or immediate constant as the second operand is determined by the srcID field. The vector arithmetic instructions (VADD, VSUB, VMLT) perform vector addition, vector subtraction, and element-wise vector product when the memory is specified as an source operand. The instructions take the first vector operand always from the vector RF, and take the second vector operand from the memory. When the second operand is specified as scalar register or immediate, then the instructions perform vector-scalar arithmetic.

Many of ML and NN techniques are composed mainly of matrix-vector multiplication. However, as more complex and computationally demanding algorithms have been emerged, most prior works implemented high dimensional operation primitives such as matrix-matrix multiplication, convolutional kernels, etc. Involving such complex primitives yet gives poor utilization efficiency especially when un-

Program 1. RNN					
LD vreg,\$0,16,\$in	%load input				
MMLT 16,16,2,shl,\$w1.ff,\$out	%feed-forward path				
LD vreg,\$0,16,\$prev.hidden	%load previous hidden layer				
VMLT mem,16,\$w1.fb	%feed-back path				
VADD mem,16,\$out	%sum ff and fb				
VADD mem,16,\$b1	%add bias				
PL vreg,\$0,16,\$tanh,11	%tanh activation				
ST vreg,\$0,16,\$prev.hidden	%save current hidden layer				
MMLT 16,16,2,shl,\$w2.ff,\$out	%output path				
LD vreg,\$0,16,\$out	%load output				
VADD mem,16,\$b2	%add bias				
PL vreg,\$0,16,\$tanh,11	%tanh activation				
ST vreg, \$0, 16, \$out	%save current output				

batched real-time processing and small sized models are required. We instead define a matrix-vector multiplication as a primitive operation to efficiently support compact fully-connected neural networks (FCNNs) or recurrent neural networks (RNNs) as they are suitable for a wide range of general sensor data classification/regression tasks [56,61–63]. The higher dimensional models such as CNNs can be linearized onto matrix-vector operation if needed. The matrix-vector multiply instruction (MMLT) performs multi-cycles of multiply-and-accumulation (MAC) using parallel SIMD datapath. It takes input vector from vector RF, and weight matrix from on-chip memory with variable sizes determined by input vector length (iLen) and output vector length (oLen). In the matrix-vector multiplication, the input vector is reused oLen times after once loaded into vector RF to minimize memory access energy. On the other hand, there is no data locality for the weight matrix and thus it is directly read from memory addressed by **\$src**. This direct access of weight matrix from scratchpad memory provides more flexibility on the matrix size, avoiding any constraints from the limited number of registers. To maximize the input vector reuse opportunity under the limited registers, the output of multiplication is directly stored back to the memory at dst. In contrast to prior works on custom ISA [55, 56, 60, 64], we include shifting capability for every output of matrix-vector multiplication so that the models can have dynamic fixed-point layer-by-layer. The shl/shr field determines the shifting direction, and the shb field specifies the amount of bits shifting.

The non-linearity have been typically used in the most of NN algorithms (activation functions) such as sigmoid, hyperbolic tangent, and rectified linear (ReLU) functions. To realize the non-linear functions, [60] defined vector-exponential and vector-div-vector instructions. However, the plain computation of exponential and division requires complex hardware and high energy cost. For the resource constrained applications, [57] implemented hard-wired ReLU function unit while scarifying the flexibility on the function types. By leveraging the error tolerant nature of NN models, the authors in [55] exploited piece-wise linear (PWL) approximation to reduce complexity and latency. Even though their LUT based implementation provides flexibility on the choice of function types, fixed and small size of LUT limits the number of function types and PWL segments. Moreover, their datapath is specifically designed to compute the activation function of NN layers. In the general sensor applications, however, there is a need for other non-linear processing tasks such as taking logarithm on the acquired audio stream. We instead define a dedicated PWL instruction (PL) for general usage. It takes either scalar or vector as operand specified by regID, \$src1, and len. In contrast to the LUT, we store the PWL parameters in the scratchpad memory. The compiler manages the memory space for PWL parameters and NN parameters to be separated. This allows more flexibility on the selection of non-linear functions to be used not only for the NN algorithms, but also for pre/post-processing of data. Multiple non-linear functions can be stored in different locations and addressed by \$src2 in the instruction. The plshb is a shift amount of input operand to determine the segment to which it belongs, and thus depends on the total number of segments for a specific non-linear function approximation.

The element-wise compute instruction (EC) implements special functions that are useful to various vector processing. The hardware FSM performs finding min/max

Program 2. Moving Average	
store.output:	
ST vreg,\$0,2,\$out	%save NN output
addr.increment:	
LD sreg, \$0,1,\$store.output	%load ST inst.
SADD imm, $\#1$	%increment ST addr.
CBR $0, \#$ out+5,addr.reset	%branch to addr.reset
ST sreg, \$0,1, \$store.output	%save back ST inst.
JMP mvavg.compute	%jump to mvavg
addr.reset:	
SSUB imm, $\#5$	%decrement ST addr.
ST sreg,\$0,1,\$store.output	%save back ST inst.
mvavg.compute:	
LD vreg,\$0,2,\$out	%load 1st sample
VADD mem,2,\$out+1	%add 2nd sample
VADD mem,2,\$out+2	%add 3rd sample
VADD mem,2,\$out+3	%add 4th sample
VADD mem,2,\$out+4	%add 5th sample
VMLT imm,2, $\#0.2$	%divide by 5
ST vreg, \$0,2, \$mvavg.out	%save mvavg output

value or index of min/max in a vector, and summing up over vector elements. The output is stored into a scalar register at \$dst. The conditional branch (CBR), direct jump (JMP), and no operation (NOP) instructions are used for program control flow.

The Program 1 shows an example assembly code for RNN computation. Input layer, hidden layer, and output layer all consist of 16 neurons. Note that since the RNN has a feedback path, additional processing steps (VMLT and VADD) are needed. Also, the hidden layer results are saved to memory at **\$prev.hidden** to be used at the next time step. Compared with [55], in which total 123-Byte of instruction memory is consumed for even smaller sized RNN, this work takes only 52-Byte of memory. Although the ISA in [60] also consumes 64-Byte of instructions, it only implements FCNN which typically requires smaller number of instructions. Moreover, while >130 instruction fetches occur during one iteration of RNN computation in [55] , the proposed ISA needs only 13 instruction fetches, mitigating the fetch/decode energy burden. Besides NN-specific operations, the proposed ISA can be used to

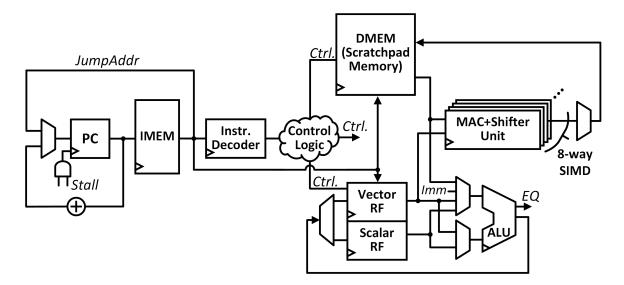


Figure 3.2: The microarchitecture of on-sensor NN processor.

realize basic data processing tasks. The Program 2 implements 5-sample moving average on the NN output. By using branch, jump and scalar operations, the NN output is stored to different locations in memory for each time step, and retrieved to compute moving average. Supporting simple algorithms beyond NN allows deploying the processor in more wide range of sensor applications.

#### 3.2.2 Microarchitecture and Implementation

The microarchitecture of the proposed on-sensor NN processor is shown in Figure 3.2. It is a five-stage 16-bit processor with additional SIMD datapath to process arbitrary NNs and general-purpose programs. It consists of program counter (PC), instruction memory (IMEM), instruction decoder, control logic, data memory (DMEM), scalar and vector RF, ALU, and 8-way SIMD datapath. The IMEM and DMEM are implemented by the low-leakage 7T SRAM cell, which is introduced in Section 3.2.3. In the standby mode, both IMEM and DMEM retain the loaded program and data while all other circuits including the memory peripherals are reset and power-gated to minimize leakage current. The DMEM is used as a scratchpad memory which requires programmer to take control of all data movement to and from it by using the instructions set. All arithmetic operations are performed in the ALU except matrix-vector multiplication.

To exploit the parallel, independent computational characteristic of matrix-vector multiplication, we employ separate multi-way SIMD datapath beside the generalpurpose ALU. Each lane consists of MAC circuits followed by a shifter. To reduce both active and standby energy, the SIMD datapath has reduced-bit precision while the ALU supports full 16-bit computation. This small fixed-point data type is one of key differentiator from a software-based implementation on microprocessor. In particular, we use 4-bit per weight and 8-bit per neuron activation since the 4-bit quantization is proved to energy optimal in many cases [65, 66]. This quantization greatly reduces not only active/standby energy of computation circuits, but also onchip storage requirement. Hence, both memory access energy and retentive standby energy are minimized. The MAC circuits consist of a  $4b \times 8b$  multiplier and a 24-bit accumulator. In contrast to prior works [55, 56, 64], a dedicated shifter dynamically moves decimal point of MAC output layer-by-layer to compensate the accuracy loss from the reduced precision. Since our primary design concern is energy efficiency, not the throughput, the number of SIMD lanes is determined by the bandwidth of memory. The 8-way datapath requires 8 weights per cycle (32 bits per cycle) of memory bandwidth, which is reasonable design choice for our 32-bit-word low power SRAM macro in Section 3.2.3.

We implement 4KB IMEM, 12KB DMEM, and 32-element vector RF. Except the retentive SRAM memory cells, all circuits have header transistors for power-gating at standby mode. The header transistors are carefully sized to trade-off between the leakage current and enter/exit time of standby mode. The simulated charge and discharge time of virtual supply rails is <1.6ms, which enables sufficiently fast transition between modes especially when considering the relaxed latency requirement of most sensor applications.

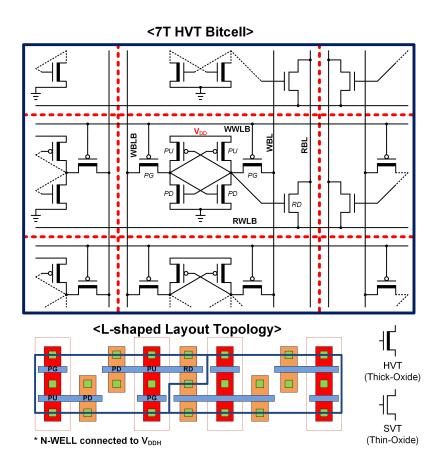


Figure 3.3: 7T HVT SRAM bitcell and layout.

#### 3.2.3 Ultra-Low Leakage 7T SRAM Memory

The NN algorithms demand large amount of data to compute. Especially, FCNNs and RNNs suffer from parameter dominance in contrast to that CNNs are typically facing computation bottleneck [67]. Since the off-chip memory access is avoided due to its energy and latency costs, realizing dense yet energy efficient on-chip memory is crucial for on-sensor NN capability. Moreover, the on-chip memory should retain all programs and parameters during standby mode, the leakage power of on-chip memory would dominate overall average power of sensor system throughout its lifetime. SRAM memory has been widely used as an on-chip storage of embedded NN accelerators and processors because of its density, speed, and cost. However, conventional 6T SRAM suffers from reduced robustness particularly for low voltage designs. As a result, 8T and even larger bitcells have been proposed for low-power applications, but came with the expense of density. 8T SRAM usually incurs  $\sim 30\%$  of area overhead [68–70]. And these issues are further complicated by the need for ultra-low leakage requirement. To achieve fW/bit of standby power, 10T HVT bitcell is proposed in [71], but the bitcell size is almost 4× larger than 6T SVT bitcell, which therefore limits the size of NN models under the area constraint of miniaturized sensor platforms. Hence, low leakage, low voltage tolerant SRAM design that also achieves reasonable area density should be exploited.

Figure 3.3 shows the proposed 7T bitcell [72], which includes 6 HVTs and a single SVT read device. The HVT devices have less leakage current by orders of magnitude than SVT devices. Moreover, a dedicated read transistor allows separate optimization on read and write path, enabling low voltage operation. However, read access to a 7T topology causes substantial short-circuit current from unselected cells. Since the typical NN workloads require large number of read accesses, this issue incurs high energy penalty to the NN processor. The proposed 7T SRAM introduces an Auto-Shut-Off mechanism in which the selected read wordline is automatically disabled during read, thereby maintaining the unselected read device as off-state. This mechanism reduces 7T read energy by  $6.8 \times$ . The 7T SRAM cell is  $2.3 \times$  smaller than the 10T bitcell in [71] while enables 3.35 fW/bit standby power. When compared with conventional 6T SVT SRAM, this 7T cell achieves >3500 × reduction in standby power.

#### 3.3 Acoustic Object Detection Sensor System

To prove the efficacy of the proposed on-sensor NN processor, we developed an acoustic sensor system for object detection. The system aims to detect different types of machines by their sounds. Figure 3.4 shows the overall system block diagram. The proposed sensor system integrates multi-stage active amplifiers, 8-bit SAR ADC, dig-

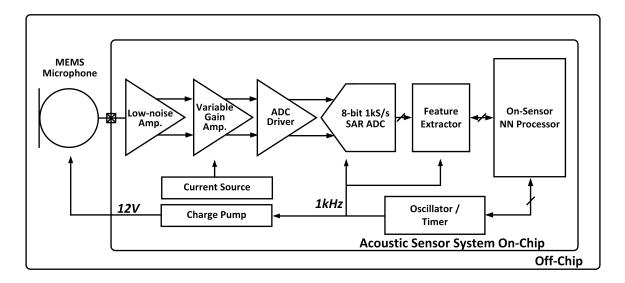


Figure 3.4: Overall block diagram of acoustic sensor system for object detection.

ital feature extractor, and the NN processor described in Section 3.2. A charge pump generates 12V of bias voltage for the passive off-chip MEMS microphone. Since target machinery sounds in the frequency domain show that acoustic features are mainly concentrated within a relatively narrow bandwidth of <500 Hz, and concentrated in a few narrow sparse tones with high-power levels, we employ the same approach introduced in [42] for the amplifiers, ADC, and feature extractor to reduce overall power consumption of the sensor system. The signal characteristic allows the system operates with a relatively low SNR and bandwidth, which reduces the burden on the amplifier noise performance, and thus current consumption. In contrast to [42], we inserted an ADC driver between the variable gain amplifier and ADC to help signal settling at ADC sampling capacitor. The ADC driver also reduces current consumption of variable gain amplifier further by the relaxed settling requirement. The tone-of-interest (ToI) DFT feature extractor proposed by [42] extracts frequency features tone-by-tone from stationary machine sounds, minimizing instantaneous power consumption. We implement the feature extractor with more programmability on the parameters than [42]. The supported DFT sizes are 192-, 256-, 384-, 512-, and 1024-point, and total 4, 8, 16, 20, or 32 of features can be extracted during a frame.

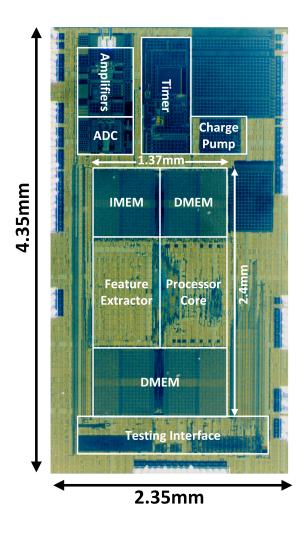


Figure 3.5: Die photograph of acoustic sensor system with on-sensor NN processor. Extracted feature data is transferred to the DMEM of NN processor by a system bus. An on-chip oscillator produces 1 kHz of clock for analog front-end, ADC, and feature

extractor, while the NN processor operates with separate own clock. A low power on-chip timer is also integrated for the duty-cycled operation of sensor system.

## 3.4 Measurement Results

The proposed on-sensor NN processor and integrated sensor system is fabricated in 180nm CMOS process as shown in Figure 3.5. The area of acoustic sensor is 10.2 mm<sup>2</sup>, and the NN processor occupies 2.6 mm<sup>2</sup>. The proposed on-sensor NN processor

	This Work	[55]	[56]	[57]	[73]	[61]
Process	180nm	32nm	130nm	28nm	65 nm	40nm
$Area(mm^2)$	2.6	2.23	0.38	5.76	1.04	7.1
NN types	FCNN/RNN	FCNN/RNN	FCNN	FCNN	RNN	FCNN
Memory	16KB	N/A	$2.1 \mathrm{KB}$	1MB	32KB	$270 \mathrm{KB}$
Voltage	$0.6\mathrm{V}/0.95\mathrm{V}^{1}$	1.6V	N/A	$0.715\mathrm{V}$	$0.575\mathrm{V}$	$0.65\mathrm{V}$
Clock	1MHz	$74 \mathrm{MHz}$	4MHz	$667 \mathrm{MHz}$	$250 \mathrm{kHz}$	$3.9 \mathrm{MHz}$
Energy efficiency	$400 \mathrm{GOPS/W}$	$59 \mathrm{GOPS/W}$	$37 \mathrm{GOPS/W}$	$264 \mathrm{GOPS/W}$	$270 \mathrm{GOPS/W}$	$187 \mathrm{GOPS}/\mathrm{W}^2$
Active power	$20\mu W$	$1.25 \mathrm{mW}$	$153\mu W$	$20.3 \mathrm{mW}$	$5\mu W$	$288\mu W$
Standby power	$440 \mathrm{pW}$	N/A	N/A	$3 \mathrm{mW}$	614 nW	$>20\mu W$
Architecture	Processor	Processor	Processor	Hard-wired	Hard-wired	Processor
Custom ISA	Yes	Yes	Yes	No	No	No

 Table 3.2: Comparison of Neural Network Processor

<sup>1</sup>Supply voltage for SRAM memory.

<sup>2</sup>Re-calculate the number as OPs=MACs.

consumes  $20\mu W$  at 1MHz with 0.6V of supply for core and 0.6V/0.95V dual supply for SRAM. The 8-MOPS of peak throughput is achieved, which is suitable for smallto-medium sized FCNN or RNN of sensor applications. We compare our processor with prior state-of-the-arts in Table 3.2. Although there exist NN accelerators having >TOPS/W of energy efficiency, their target applications are large, deep networks that can benefit from higher parallelism. Here, we compare with more compact implementations of modest sized NN models for on- or near-sensor applications. The [57] shows  $\sim$ 5.3-TOPS of peak performance, but >20mW of power consumption is not suitable for millimeter-scale power source. The RNN accelerator in [73] consumes only  $5\mu W$ of power which is lower than our processor. However, their standby leakage power is >600 nW, translated to only several days of lifetime on millimeter-scale  $16\mu$ Ah battery [47]. Moreover, both designs have hard-wired accelerators that can not be used beyond fixed NN topology. The [55] and [56] support custom ISA with processor-type architecture, allowing more flexibility to end-users. However, their energy efficiency is below <100-GOPS/W due to the absence of data reuse and parallelism. Moreover, none of the prior works consider the standby power, resulting in high leakage current at idle state. The proposed on-sensor NN processor shows 400-GOPS/W of energy efficiency and 440pW of standby power with custom ISA support, enabling built-in

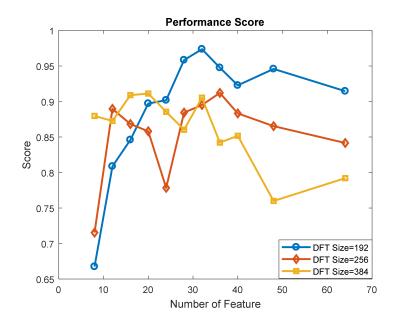


Figure 3.6: Algorithm parameter sensitivity analysis.

NN capability on sensor node.

The implemented acoustic sensor system is also measured as follows: The amplifiers operates with 1.2V supply voltage and the gain varies from 31- to 60-dB with 430Hz of 3-dB bandwidth. The integrated input referred noise is measured as  $17\mu V_{rms}$ , consuming 4.9nW. Measured signal-to-noise and distortion ratio (SNDR) of the ADC is 47.02dB, which corresponds to 7.5bit effective number of bits (ENOB). The ADC operates with 0.6V supply at a sampling rate of 1kS/s. The charge pump consumes 4.7nW operating at 1kHz to generate 12V bias voltage for microphone. The ToI DFT feature extractor is implemented with HVT devices and always-on with 0.6V supply, consuming 3.6nW.

To evaluate acoustic object detection performance, we use DARPA-provided dataset and perform a task that distinguish between generator, car, truck, wind, and quiet by sounds under quiet, rural, and urban background noise. The NN is trained with MATLAB software. We use mutually exclusive dataset for training and testing. The algorithm parameters such as ToI DFT size, number of features, NN configuration affect both accuracy and power. Since the ToI DFT is serialized feature extraction

Showed data Target	Generator	Car	Truck	Quiet	Wind
Generator	100.00%	0.07%	0.00%	1.60%	0.00%
Car	0.00%	98.26%	0.17%	2.79%	0.20%
Truck	0.00%	0.11%	98.41%	3.05%	0.07%
Quiet	0.00%	1.40%	1.37%	91.49%	0.10%
Wind	0.00%	0.16%	0.05%	1.07%	99.63%

Table 3.3: Measured Acoustic Object Detection Accuracy

method, the DFT size and the number of features determine a frame length. Larger DFT size and number of features give higher frequency resolution and more spectral information to the NN, respectively, but not necessarily always helpful. For example, if we select larger number of features or DFT sizes, a frame length becomes longer and thus, the NN only sees fewer frames to make a decision at fixed inference/s rate requirement. Therefore, we first analyze parameter dependency on the performance to get the best accuracy results. We define a performance score as following:

performance score = 
$$\sqrt{PD_{worst} \cdot (1 - FA_{worst})}$$
 (3.1)

where PD is positive detection rate and FA is false alarm. First, we evaluate the score depending on the DFT sizes and the number of features at a same fixed NN topology, as shown in Figure 3.6. The best accuracy is achieved when DFT size = 192 and # of features = 32. By using these parameters, we next evaluate the score for different NN configurations. Figure 3.7 shows the final NN topology we use for the acoustic object detection task. Total 5 of 6s-frame (192ms  $\times$  32) are tied as one feature vector, realizing 160 neurons of input layer. Features are normalized for each inference. The NN has 2 hidden layers with 80 and 32 neurons, and 5 output neurons for 5 acoustic objects, consuming 7.65KB of DMEM for weights. The sigmoid function is used as non-linear activation function within the layers, and the final probability for each object is calculated by softmax function. The 5-sample moving average is

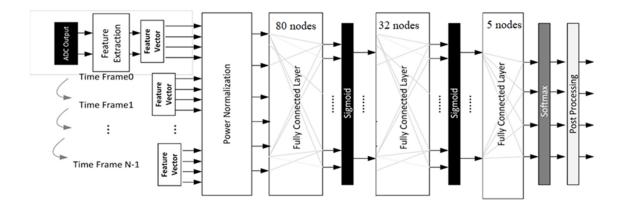


Figure 3.7: Neural network topology for acoustic object detection.

performed as post-processing at the last to further smooth the results. All these processes after feature extraction are programmed by the proposed custom ISA, and the program is loaded into the IMEM of NN processor (total 476-Byte). Table 3.3 shows the measured accuracy. For all targets under three types of ambient noise, the sensor achieves >90% of positive detection rate, and <3% of false alarm rate.

Since the decision interval is 30s (5-frame  $\times$  6s/frame) and the throughput requirement is ~520-OPS, the on-sensor NN processor is duty-cycled as shown in Figure 3.1b, benefited from ultra-low standby power. The average power consumption of the NN processor is 33nW when executing the aforementioned algorithm. Figure 3.8 shows the overall power breakdown of acoustic object detection sensor system.

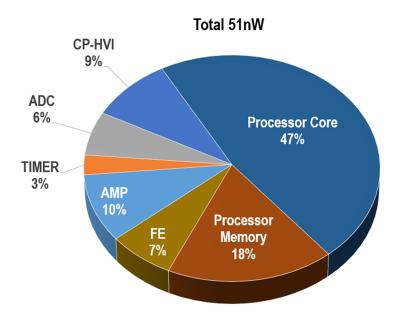


Figure 3.8: Power breakdown of the proposed acoustic sensor system.

## CHAPTER IV

## An Acoustic Signal Processing Chip With Nanowatt Power Voice Activity Detection

## 4.1 Introduction

Voice user interfaces are widely adopted in various devices as the human voice is one of the most natural and information-rich interfaces between humans and machines. Minimizing the power consumption of voice processing is particularly crucial to meet power budgets when the system becomes smaller as the battery size imposes severe power and energy constraints on system design [47]. In many practical applications, acoustic events-of-interest occur infrequently. Constant listening and detection of keywords are very powerhungry. Instead, the use of an always-on voice activity detector (VAD) as a system wakeup mechanism is a popular alternative [62, 74–78], and subsequent power-hungry processing is enabled by the VAD to save overall system power, as shown in Figure 4.1. The acoustic wakeup detector consumes much less power than constant listening for keywords since it only detects whether an incoming signal contains a human voice. However, since the events occur infrequently, the always-on acoustic wakeup detector typically dominates the system power consumption, and therefore, minimizing the VAD power consumption itself is a critical design challenge.

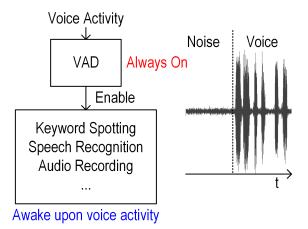


Figure 4.1: Always-on voice activity detection as a wakeup mechanism. Advanced processing is enabled upon voice activity detection to save the overall power.

A previous acoustic wakeup detector [42] consumes just 12 nW but it is specifically designed to detect "stationary" events whose signal features are invariant over a relatively long time (a few seconds) and very narrow in frequency (2-Hz bandwidth or 0.5-s extraction time for each feature). The approach in [42] is not applicable to a nonstationary target, such as voice activity containing time-varying features that need to be extracted with a short (tens of ms) interval. Prior VAD chips [62,74] demonstrated reliable performance but consumed significant power (>20  $\mu$ W) and lacked an analog front end (AFE), which would further increase the power. More recent analogdomain feature extraction-based VAD chips [75, 76] also reported  $\mu$ W-level power consumption, and their simple decision tree [75] or fixed neural network (NN)-based approach [76] limited broader use for various acoustic event targets. Moreover, the VAD chips [62,74–76] were tested using only electric analog audio signals, rather than actual audio signals. Therefore, additional components and their power consumption

Typical VADs consist of two parts [79–81]. A feature extractor which converts the incoming signal into low-dimensional but dense acoustic features, and a classifier that takes a feature set input and produces a binary decision: Speech or non-speech. Both design of feature extractor and classifier significantly affect overall system power, accuracy, latency, and scalability.

The main challenge in reducing the overall VAD power is to reduce the power required for feature extraction since it is typically computation-intensive and operates continuously without duty cycling. Conventional approaches [62, 74, 81] used digital fast Fourier transform (FFT)-based feature extraction, yet FFT itself consumes >2  $\mu$ W even with extensively relaxed throughput/latency constraints [74]. To reduce power consumption, [75, 76] exploited analog-domain feature extraction techniques. However, the parallel filter bank at the voice-band is still the most power-hungry block, preventing sub- $\mu$ W operation. Instead of using parallel feature extraction, such as an analog filter bank or digital FFT, a serialized discrete Fourier transform (DFT) on tones-of-interest approach was introduced in our previous work [42] for low-frequency (<500 Hz) signal targets. However, applying the same technique to the voice-band (up to 4 kHz) frequency significantly increases the power consumption of both the AFE and the digital feature extractor proportionally with signal bandwidth, limiting the usefulness of this technique.

To improve the accuracy and scalability of the VAD system, the NN-based classifiers have been recently proposed [82–86]. Compared to other machine learning classifiers, such as decision tree [75] or support vector machine (SVM) [42], NN-based classifier have shown the improved performance [87], immunity to difficult noise scenarios [88], and strong scalability to multiple acoustic targets [89] and large-scale corpora [90], becoming a strong candidate for real-world applications.

This chapter presents a programmable acoustic signal processing system for both VAD and non-voice acoustic event detection based on NN classifier. We use a mixerbased architecture that sequentially scans and down-converts the 4-kHz bandwidth signal to a  $\leq$ 500-Hz passband, reducing amplifier, analog-to-digital converter (ADC), and digital signal processor (DSP) power by 4×. The NN processor employs computational sprinting, which minimizes static energy dominance in low frequency/voltage regimes, providing  $12 \times$  power reduction in the digital domain. In addition to a VAD, the system features an inaudible acoustic signature detection mode to enable remote silent system wakeup. The proposed always-on VAD consumes 142 nW, which is  $8 \times$  lower than that reported in the literature for state-of-the-art works. In this chapter, Section 4.2 describes the overview of the VAD system. Sections 4.3 and 4.4 show its circuit implementation, Section 4.5 introduces acoustic signature detection, and Section 4.6 discusses the measurement results. Finally, Section 4.7 concludes this chapter.

#### 4.2 VAD System Overview

Figure 4.2a shows the overall system architecture with two signal chains: an always-on ultra-low power (ULP) chain and a high performance (HP) chain that wakes upon event detection by the ULP chain. The system has two modes based on the two chains: a 142-nW ULP mode and an  $18-\mu W$  HP mode. The HP chain is power-gated in the ULP mode, while the ULP chain is always on. When a target event is detected in ULP mode, an off-chip microprocessor ( $\mu$ P) activates the HP mode, which enables more powerful feature extraction and classification to complete additional complex tasks at the cost of power consumption. The HP mode also supports real-time audio compressing and streaming to off-chip eFlash for general purpose post-processing [47]. The HP chain consists of 4-kHz bandwidth and 8-kS/s sampling rate with a conventional AFE architecture consisting of a low-noise amplifier (LNA), programmable amplifier (PGA), ADC driver (DRV), and ADC. In contrast, the ULP chain employs a digitally controlled mixer between the LNA and PGA to shift the desired signal frequency down to 500-Hz bandwidth to lower the Nyquist rate to 1 kS/s after the PGA. Both the ULP and HP chains share the same NN processor, but it operates with a different power scale and network model for each mode. An

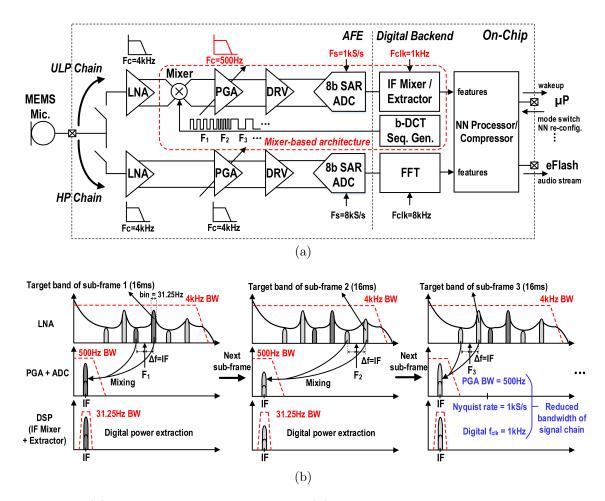


Figure 4.2: (a) VAD system architecture. (b) Operating principle of mixer-based sequential frequency scanning.

external clock from a 32-kHz crystal oscillator is divided into 8 and 1 kHz for the HP and ULP chain, respectively.

Figure 4.2b shows the mixer-based sequential frequency scanning operation that reduces AFE and DSP power consumption in the ULP mode by lowering their bandwidth and sampling rate to 500 Hz and 1 kHz, respectively. The incoming signal from the microphone is amplified by an LNA with the full 4-kHz bandwidth. At this point, the mixer, switched by a binary discrete cosine transform (DCT) sequence, immediately down-converts the frequency of the desired feature to a programmable intermediate frequency (IF) of <500 Hz. The digital binary sequence generator supports an arbitrary DCT frequency for the mixer switch control; for example, the

4-kHz band can be divided into 31.25-Hz frequency bins using a 128-pt DCT, and the energy content of 32 bands out of 128 is sequentially extracted by sweeping the DCT frequencies  $(F_1, F_2, ..., F_{32})$ . The 32 bands are chosen during NN training for each target event. The IF down-converted signal is further amplified and low-pass filtered with 500-Hz bandwidth (via a PGA) and digitized at 1 kS/s. Finally, the digital IF quadrature mixer down-converts the signal to dc, and feature power is measured. With a DCT length of 16 ms per feature (128-pt DCT with 8 kHz binary mixing), 32-feature extraction requires a 512-ms frame. The mixer-based structure reduces the bandwidth, sampling rate, and clock frequency of the AFE and DSP after the mixer; thus, the feature extraction power consumption is decreased from 225 nW (simulation; based on LNA and PGA at 4 kHz of bandwidth, DRV and ADC at 8 kS/s of sampling rate, and digital FFT at 8 kHz of clock) to 60 nW (measured; including LNA, PGA, DRV, ADC, IF mixer/extractor, and binary-DCT sequence generator). The programmable IF is set to  $\approx 250$  Hz to reduce the PGA 1/f noise effect while the image aliasing issue of non-quadrature mixing and imperfection of first-order filtering is mitigated (without noticeable event detection accuracy degradation) by an NN trained with the image-aliased and attenuated signals.

### 4.3 Analog Front-End Implementation

#### 4.3.1 Overall Architecture

Figure 4.3 shows the AFE circuit diagram with ULP and HP chains. Both chains share a single MEMS capacitive microphone and a charge pump. Depending on the operation mode, the chain selection switches select one chain. The HP chain consists of a 31.3-dB gain LNA, 4.6–31.3-dB gain PGA, 8-bit ADC, and an ADC DRV. The ULP chain also includes all the blocks of the HP chain but operates with lower power consumption as it targets relaxed noise performance and ULP operation. Moreover,

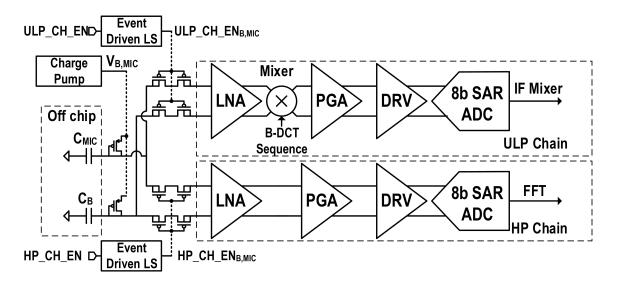


Figure 4.3: AFE block diagram with ULP and HP chains.

the ULP chain has a mixer between the LNA and PGA. The mixer is a passive mixer similar to a typical chopper and is controlled by the binary DCT sequence generator.

#### 4.3.2 Charge Pump and 10-V Level Shifter

Microphone sensitivity is proportional to the microphone bias voltage, and therefore, we use a three-stage Dickson charge pump to generate 10-V bias [47]. Because the MEMS microphone is capacitive, the charge pump only needs to drive negligible loads. The charge pump uses the 8-kHz clock to minimize possible clock signal coupling to the signal chain (4-kHz BW) and consumes only 13 nW (measured). The diode connected PMOS sets the corner frequency of the voltage bias to be well below the microphone response range (<75 Hz) to avoid altering the acoustic response in the system. CB is an external capacitor to match the input impedance.

To switch the modes between ULP and HP, level shifters shift the control signal voltage level from nominal VDD to 10 V, since the LNA inputs see signals in the 10-V domain. Figure 4.4 shows the proposed level shifter. Because 10 V is much higher than the transistor oxide breakdown, coupling capacitors implemented with a metal–oxide–metal (MOM) structure are used to bridge to the high voltage in the

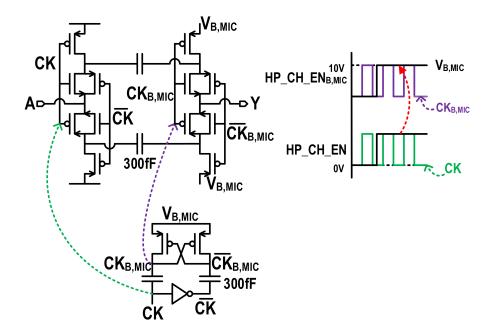


Figure 4.4: 10-V level shifter shifts up nominal VDD level to 10 V with periodic refresh. Its waveforms are shown at right.

level shifter. However, the coupling capacitors may suffer from transistor leakage due to infrequent mode switches. To avoid leakage, the capacitors are periodically refreshed with the clock. It is complementarily switched for continuous operation.

#### 4.3.3 Low-Noise Amplifier and Programmable-Gain Amplifier

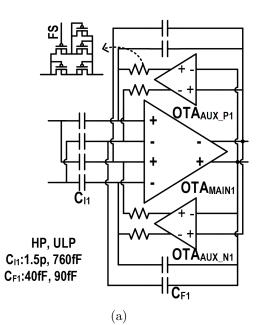
The first stage amplifier determines the overall noise performance of the analog signal chain as it is the most noise-sensitive block in the system. Figure 4.5a shows the block diagram of the proposed LNA. It uses capacitive feedback and pseudo-resistor dc-servo loops for low-power and small area implementation, respectively. LNA gain is set by the ratios of  $C_{I1}$  to  $C_{F1}$ . The ULP LNA gain is 18 dB, while the HP LNA gain is set to 31.3 dB to detect smaller acoustic signals. Figure 4.5b shows the main operational transconductance amplifier (OTA) with common-mode feedback (CMFB). A conventional differential difference amplifier (DDA)-based common-mode feedback shows poor linearity when the signal is large, as shown in Figure 4.6 (red line) [91,92]. To enhance the output range and linearity, we use two different loops for

the CMFB. One employs coupling capacitors for high bandwidth and good linearity across the signal amplitude. The other loop uses a DDA with a pseudo-resistor and is only responsible for setting the dc level.

The main OTA adopts an inverter-based cascode amplifier for better noise efficiency [42, 91]. PMOS and NMOS input transistor pairs are separately biased, and hence they have two pairs of  $C_{I1}$  and  $C_{F1}$  and also have two dc-servo loops. The sizes of the input transistor pairs are determined for balanced 1/f noise and thermal noise. The auxiliary amplifiers (auxamp) in the dc-servo loops shift the output commonmode voltage of the main OTA to an optimal bias point for each PMOS/NMOS input pair to maximize the LNA output range. The implementation of the aux-amp is shown in Figure 4.5c. Very high resistance (>T $\Omega$ ) can be readily achieved with a pseudo resistor in a small area, but its resistance varies substantially and is nonlinear when the voltage difference between the two terminals is large. In particular, mismatch among parasitic diodes and intrinsic gate diodes causes amplitude-dependent drift that may cause amplifier saturation. The aux-amps attenuate the maximum amplitude seen by the pseudo-resistors and hence, improve the operation range and linearity.

Figure 4.7 shows the PGA implementation. Since PGA is less sensitive to noise than LNA, the PGAs main OTA (OTA<sub>MAIN2</sub>) uses only a PMOS input pair for the maximum output range. The gain is adjustable between 4.6 and 31.3 dB by changing  $C_{I2}$  for both the ULP and HP chains.  $C_{L2}$  sets 500-Hz BW and 4-kHz BW for the ULP and HP chains, respectively.

Typical audio systems activated by a VAD could experience front end clipping (FEC), which may result in losing the first portion of each audio segment in passing from noise to voice activity due to the transition time between modes [93]. This effect is exacerbated especially in low power systems with pseudo-resistors since their extremely high resistance makes the settling time exceedingly long. In this design,



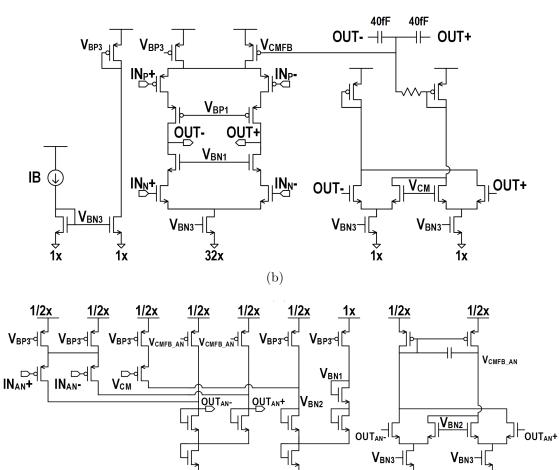


Figure 4.5: (a) LNA circuit diagram. (b)  $OTA_{MAIN1}$ . (c)  $OTA_{AUX_N1}$  and their bias implementation.  $OTA_{AUX_P1}$  are implemented similarly with the opposite type of transistors.

(c)

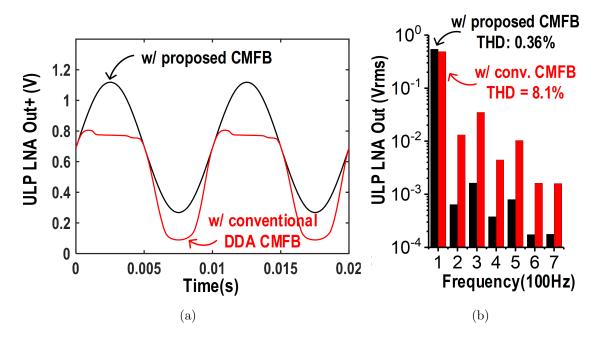


Figure 4.6: (a) ULP LNA output waveform with conventional DDA CMFB (red) versus with proposed CMFB consisting of coupling capacitors and DDA (black). (b) Its spectrum (simulated).

we minimize the ULP-HP transition time by temporarily turning on fast settling switches [see Figure 4.5a] during the transition. Figure 4.8 shows the measured results. The common-mode voltage settling time is reduced from 6 s to 100 ms, proving the effectiveness of this method.

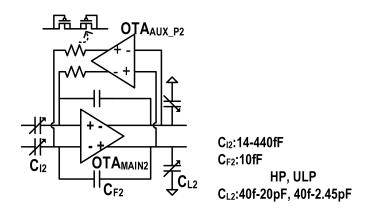


Figure 4.7: PGA circuit diagram.

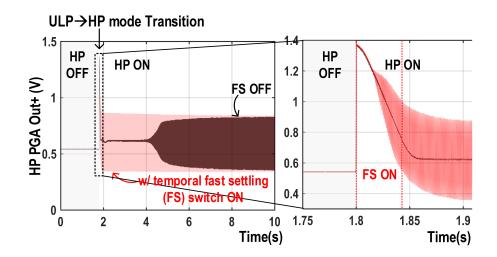


Figure 4.8: Measured HP PGA output showing ULP-HP mode transition time. By turning on fast settling switches for 30 ms, the settling time reduces from 6 s (black) to 100 ms (red).

# 4.4 Digital Back-End Implementation

### 4.4.1 Overall Architecture

Figure 4.9 shows the digital back-end architecture. In the ULP mode, while the binary DCT mixer sequence generator produces a square wave for the mixer at the AFE, the IF mixer and extractor receive the ADC output to down-convert an IF signal into dc and to extract signal power as features for NN classification. Once a set of features is collected at the feature buffer during a frame, it is transferred to the NN processor as an input via a bus shared among digital blocks. A linear feedback shift register (LFSR) replaces the binary DCT mixer sequence generator in acoustic signature detection mode, as explained in detail in Section V. In HP mode, the first-in first-out (FIFO) buffer performs the windowing of the ADC samples for both compressions [47] and FFT. The NN processor in HP mode computes the FFT and classification. The always-on ULP modules are implemented with thick oxide I/O devices to suppress leakage, while power-gated HP modules including the NN processor are designed with standard devices. Due to the mixer-based architecture, digital

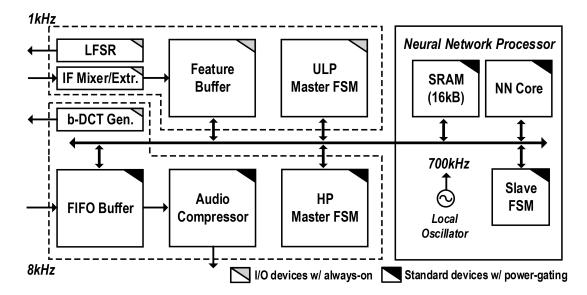


Figure 4.9: Digital backend architecture.

processing after the ADC in ULP mode runs at 1 kHz rather than the 8-kHz Nyquist rate, yielding 41% reduction in digital feature extraction power consumption.While the binary DCT mixer sequence generator runs at 8 kHz, it only consumes 4 nW.

#### 4.4.2 Binary DCT Mixer Sequence Generator

In ULP mode, the binary DCT mixer sequence generator shown in Figure 4.10 controls the feature frequency band selection by creating a DCT basis waveform to be correlated with the incoming signal. The circuit accumulates a programmable phase, which is expressed as follows:

$$\Delta \theta = \frac{\pi}{2N}k, \quad 0 \le k \le N - 1 \tag{4.1}$$

where N is the DCT size and k is the index of selected scanning frequency bands (i.e.,  $F_k$ ), either as is or doubled by shifting to generate a DCT basis function by the following equation:

$$\cos(\theta) = \cos(\Delta\theta(2n+1)), \quad n = 0, 1, ..., N-1$$
 (4.2)

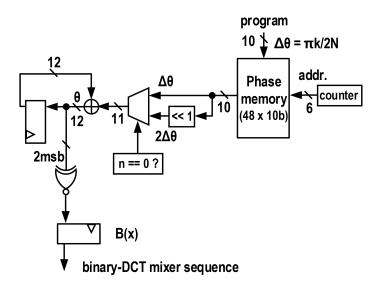


Figure 4.10: Binary DCT mixer sequence generator circuits.

where n is the accumulation step. By simply using 2 MSBs of the accumulated phase instead of the exact cosine calculation, the binarized DCT basis waveform can be obtained by the following equation:

$$B(f(k,n)) = B(\cos(\frac{\pi}{2N}k(2n+1)))$$
$$B(x) = \begin{cases} 1, & x > 0\\ -1, & x \le 0 \end{cases}$$
(4.3)

The DCT size N determines the resolution of the frequency bins and frame length, and the number of selected feature frequencies, m, is specified by the number of different accumulation phase values (i.e., the number of different k values). The k values are arbitrarily programmable to set particular scanning frequencies and determined during the NN training process. This design supports N = 32, 64, ..., 1024, and m = 16, 20, 32, 48.

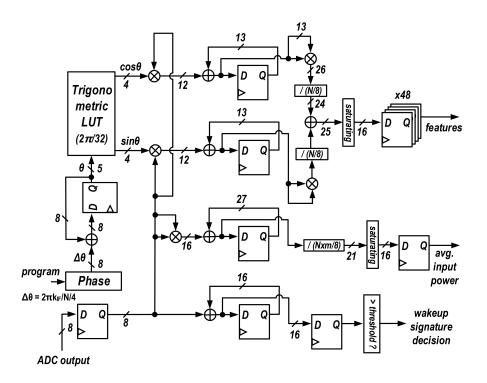


Figure 4.11: IF mixer and extractor circuits.

#### 4.4.3 IF Mixer and Extractor

Figure 4.11 shows the IF mixer and extractor that perform quadrature mixing of the IF signal from the ADC and calculates the power as a scanned frequency feature. The extracted feature can be expressed as follows:

feature = 
$$log((|X[k]|)^2),$$
  

$$X[k] = \frac{1}{\sqrt{N/8}} \sum_{n=0}^{N/8-1} \left(\frac{1}{8} \sum_{i=8n}^{8n+7} B(f(k,i))x[i]\right) \cdot e^{-j\frac{2\pi}{N/4}k_{IF}n}$$
(4.4)

where x is the input signal, n is the ADC sample index, and  $k_{IF}$  is the index of the IF frequency. Note that the computation inside the parentheses in Equation 4.4 is done in the AFE by the mixer and the low-pass filter of the PGA. The 4-bit quantized cosine and sine functions are generated by the phase accumulator and lookup table. The phase value can be programmed by the index  $k_{IF}$  to set the proper IF frequency, avoiding interference such as 60-Hz noise or other possible ambient acoustic noise.

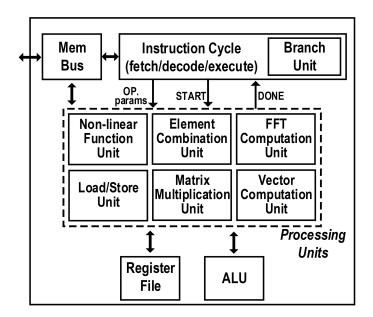


Figure 4.12: NN processor core architecture.

This circuit also computes the average input power per frame to be used for automatic gain control. Last, the ADC output is accumulated for a fixed amount of time with a separate data path that is only turned on during the acoustic signature detection mode, as explained in Section 4.5.

#### 4.4.4 Neural Network Processor

The ULP NN processor shown in Figure 4.12 employs a custom-built instruction set including matrix-vector multiplication, FFT, conditional branch, element-wise vector operation, non-linear activation, and min/max/averaging to support arbitrary network models and various pre/post-processing, as detailed in Chapter III.

The processor has 16 kB of on-chip SRAM storage (see Figure 4.9) for model parameters (4 bit per weight) and instructions. By leveraging the custom-designed high-Vth SRAM cells, the power-gated sleep retention power of the processor is only 440 pW. However, the active state leakage power is >800 nW because the processor core and SRAM peripherals consist of standard-Vth devices to meet the performance requirement of the HP mode, and this active leakage power is much higher than the

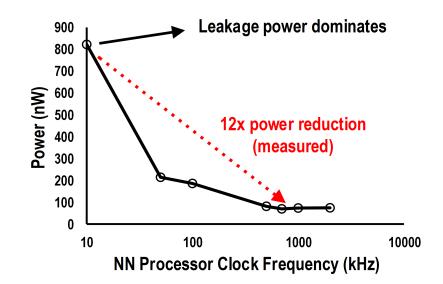


Figure 4.13: Measured power reduction from computational sprinting.

power consumed by ULP feature extraction. Hence, if the processor runs at a slow clock frequency of 1 kHz with the rest of the ULP digital processing modules to minimize dynamic power, then system power consumption would be dominated by NN processor leakage. To suppress this active leakage power, the concept of computational sprinting is adopted, minimizing the active time of the NN processor. Since ULP feature extraction operates sequentially, there is a long interval between classifications of a frame. The NN processor sprints at 700 kHz once the sequential feature extraction is complete and then is power-gated for the remainder of the next feature extraction. When 128-pt DCT, 32-feature, and a 32-32-16-2 NN model configuration are used, a duty cycle of 0.8% (sprint/sleep ratio) is achieved with 512 ms of frame interval, resulting in a  $12 \times$  power reduction in the NN processor compared with running it at 10 kHz without sprinting, as shown in Figure 4.13.

On the other hand, in HP mode, a 128-80-20-2 NN model configuration is used. The mixer-based sequential frequency scanning feature extraction is replaced by a parallel FFT based approach that extracts the full 128 features by performing the 256-pt FFT on a 32 ms of the frame. The HP mode operation reduces the latency of feature extraction by a factor of  $16 \times$  at the cost of 2.47- $\mu$ W power consumption (measured; for AFE and digital FFT feature extraction) compared to the ULP mixerbased sequential frequency scanning approach. The NN processor stays active running at 700 kHz without duty cycling or clock gating for the HP mode to maintain the  $124\times$  increased throughput of 371 kmacs/s, compared with 3 kmacs/s in the ULP mode. Unlike the ULP mode, the active leakage does not dominate the overall power consumption in HP mode.

## 4.5 Acoustic Signature Wakeup Detection

The system also features inaudible acoustic signature detection as an alternate wakeup mechanism. This feature enables user-command silent remote wakeup of the sensor node without disturbing other sensors or people around them, as shown in Figure 4.14a. The mixer-based architecture is reused to realize the signature detection, as depicted in Figure 4.14b. An incoming signal is mixed with a local pseudo-random sequence through the mixer in the ULP AFE, and then the (digitally) accumulated value for a full sequence is compared with a threshold to determine the existence of a signature with the circuit shown in Figure 4.11, as explained in Section 4.4.3. In this mode, a programmable LFSR running at 1 kHz replaces the binary DCT mixer sequence generator, producing a maximal length sequence (MLS) to be mixed with the input signal. The length of MLS ( $N_{MLS}$ ) is determined as  $2^{stage} - 1$ , where the stage is the number of LFSR stages. The LFSR tabs are arbitrarily programmed to allow a dedicated MLS for each sensor node, and to configure bit-stages of LFSR, exploiting the tradeoff between the minimum required SNR and detection latency.

The proposed sequence correlation with simple mixing requires exact phase alignment between the sequence from the transmitter and receiver. However, this phase alignment cannot be guaranteed because each sensor operates on unsynchronized independent clock sources. Running a full correlation at every sample to test all possible phases is computationally expensive. To mitigate this issue, we propose a time-drift synchronization scheme to realize correlation with simple mixing at low power. As shown in Figure 4.14c, the transmitter and receiver use intentionally mismatched sequence lengths of  $N_{MLS} + x_{MLS}$  and  $N_{MLS}$ , respectively. Due to the length mismatch by  $x_{MLS}$ , relative phases of two sequences drift over time and periodically align with each other at the beginning of the sequence, and the accumulated mixed-signal produces periodic peaks to trigger wakeup. The period of the peaks, or the worst detection latency, is determined by  $N_{MLS}(N_{MLS} + x_{MLS}) f_{LFSR}$ .

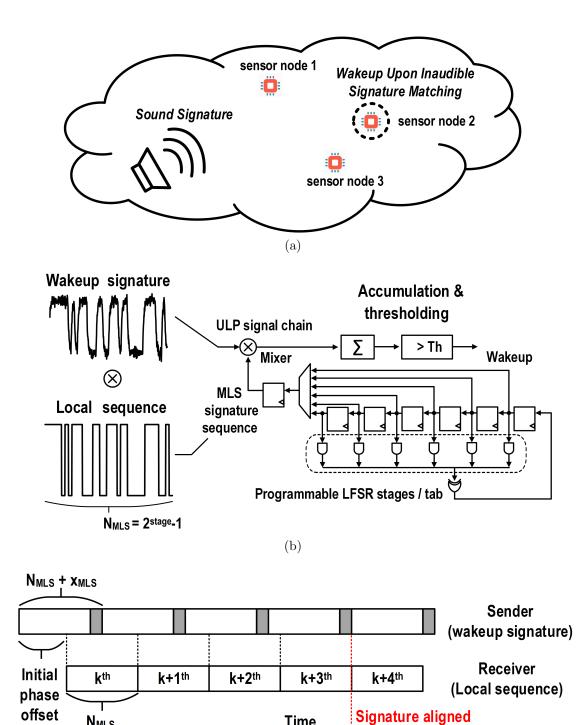
### 4.6 Measurement Results

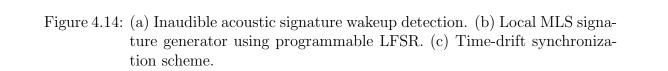
The chip was fabricated in 180-nm CMOS and integrated with a MEMS microphone, as shown in Figure 4.15. The ULP and HP chain amplifiers consumed 31 and 370 nW, respectively. The ULP chain amplifiers have 16 and 62  $\mu V_{rms}$  measured input-referred noises with the maximum and minimum gain settings, respectively, as shown in Figure 4.16a. The maximum PGA output range that satisfies 8-bit accuracy [<0.4% total harmonic distortion (THD)] is 1.45 V<sub>pp</sub>. The HP chain amplifiers have 8.7- $\mu V_{rms}$  input-referred noise across all PGA gain settings [see Figure 4.16b].

Figure 4.17 shows the measured mixer-based frequency scanning operation and input referred noise spectrum for the 64-pt DCT case. Two different applied tones, 1 and 2 kHz, were mixed down to 250 Hz in the IF, and power was extracted by DSP at two mixing frequencies each: 1) 0.75 and 1.25 kHz for 1-kHz input tone and 2) 1.75 and 2.25 kHz for 2-kHz input tone.

Figure 4.18 shows the measured ULP and HP mode power breakdown. The total ULP power was measured as 142 nW, and every block power was very balanced, which indicates a well-optimized design. The measured HP power was 18  $\mu$ W dominated by the digital circuits.

For VAD performance evaluation, 40 min of speech segments were concatenated from the LibriSpeech data set and mixed with babble noise from the NOISEX-92 data





Time

(c)

N<sub>MLS</sub>

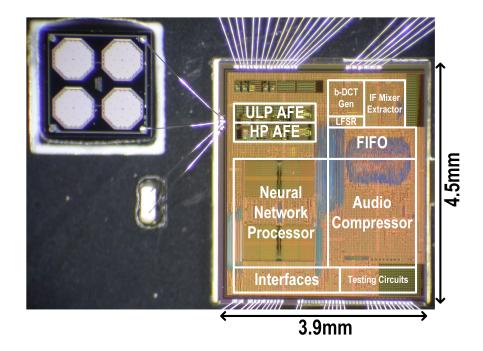


Figure 4.15: Die micrograph and system integration with MEMS microphone.

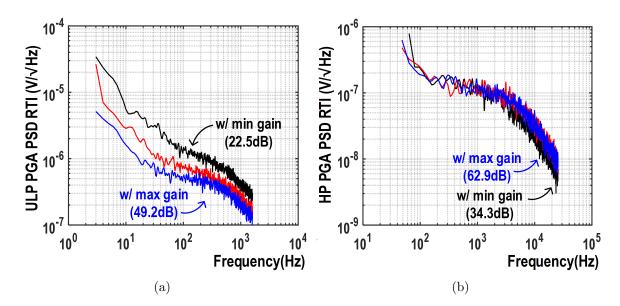


Figure 4.16: (a) ULP PGA. (b) HP PGA input referred noise spectrum density with different PGA gain settings (min, mid, and max gain).

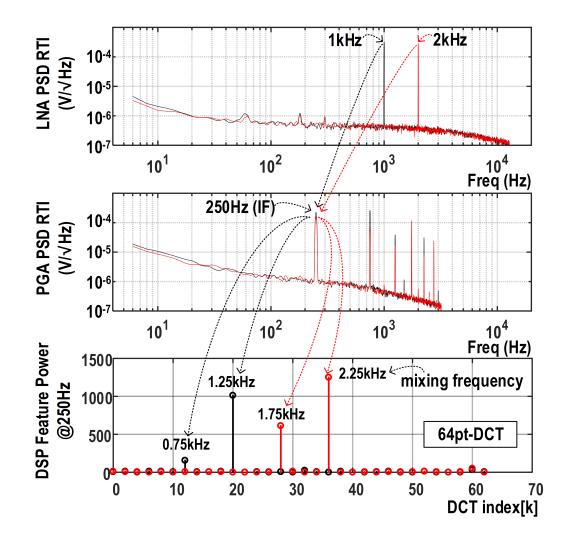


Figure 4.17: Power spectral density referred to input (PSD RTI) for LNA, PGA, and DSP. Two different applied tones (1 and 2 kHz) are mixed down to 250 Hz in IF and extracted by DSP at two mixing frequencies each (0.75 and 1.25 kHz for 1 kHz and 1.75 and 2.25 kHz for 2-kHz tone).

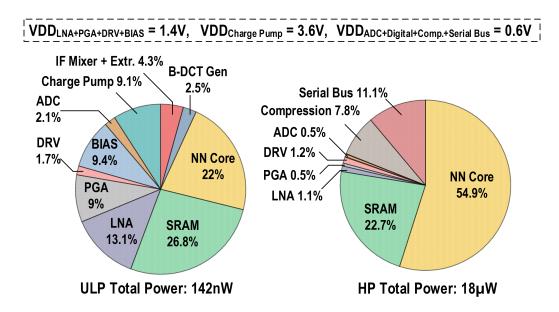


Figure 4.18: Measured power distribution of ULP mode (left) and HP mode (right).

set for training. For testing, 10 min of concatenated speech and noise segments were used. Exclusive data sets were used for NN training and evaluation to guarantee no over-fitting occurred.

We first performed electrical testing by inputting signal feeds to the LNA via an electrical connection. Figure 4.19 shows the measured receiver operating characteristic (ROC) curve with varying SNRs in the ULP mode. The detection threshold is set by the point on the ROC curve that maximizes the rectangular area formed by its coordinates. The system achieves 91.5%/90% speech/non-speech hit rates at 10-dB SNR with babble noise in the ULP mode when programmed with an NN of size 32-32-16-2 neurons with two hidden-layers, exhibiting  $\sim 7.5\%$  better hit rate with  $7 \times$  less power consumption than prior state-of-the-art works.

Unlike prior-art, we also performed an acoustic VAD test with the setup shown in Figure 4.20. The proposed chip was integrated with a MEMS microphone in the daughterboard, which includes a sound hole, and is then covered by a 3D-printed custom lid to provide an acoustic cavity for the microphone and protect electronics at the same time. Then, the daughterboard was connected to the motherboard and

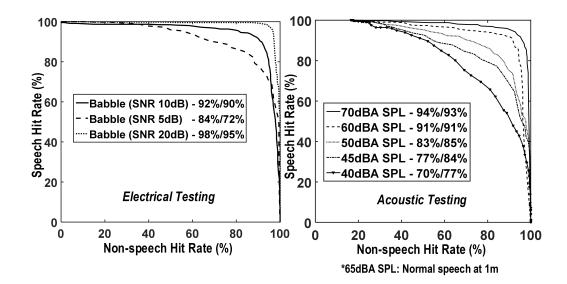


Figure 4.19: ROC curves for ULP VAD mode with varying SNRs in the electrical test (electrical connection to LNA, left) and SPLs in the acoustic test (using speaker/integrated microphone in the sound chamber, right).

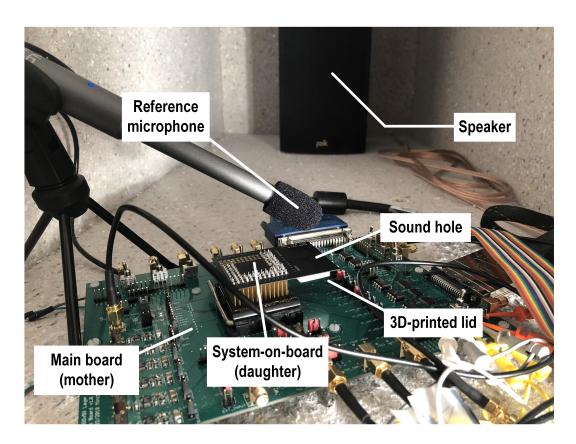


Figure 4.20: Acoustic testing setup. Proposed chip was integrated into the systemon-board with a MEMS microphone and 3-D-printed lid and tested in a sound chamber.

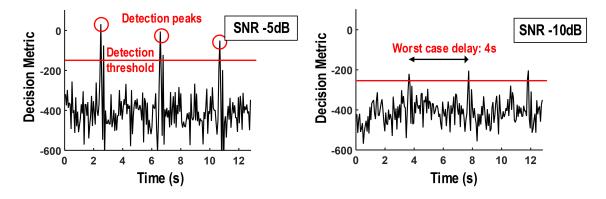


Figure 4.21: Measurement results of acoustic signature wakeup detection with MLS sequence of six stages,  $N_{MLS} = 63$ , and  $x_{MLS} = 1$  at various SNRs, showing detection down to -10-dB SNR.

placed within the sound chamber to achieve very low ambient noise, around 35-dBA sound pressure level (SPL). For acoustic testing, we concatenated speech segments without mixing any background contextual noise to measure the effect of circuit noise only. The measurement results show >83%/85% speech/non-speech hit rates with a signal level down to 50-dBA SPL, as shown in Figure 4.19. The measured AFE equivalent input noise (EIN) is 45- and 44-dB SPL (no weighting) for ULP (500-Hz BW) and HP (4-kHz BW) chains, respectively.

The measurement of acoustic signature wakeup detection was also performed. As shown in Figure 4.21, the system wakes up under exposure to as little as -10-dB SNR of white-noise-like sound when MLS signature of 6-stages,  $N_{MLS} = 63$ , and  $x_{MLS} =$ 1 is used, consuming 66 nW. The detection threshold of the decision metric is set to 10 dB to measure minimum SNR. These results prove that the system can be awoken by a signature buried in ambient noise that is inaudible to humans near the receiver. Moreover, Figure 4.22 shows that the increased stages of LFSR allow more relaxed SNR requirement at the cost of increased detection latency. Note that every added stage achieves around 3 dB of SNR gain, but pays ~4× increased latency.

Figure 4.23 shows measured logic analyzer output of overall system operation. The acoustic system stayed in the ULP mode when there was no voice. The system

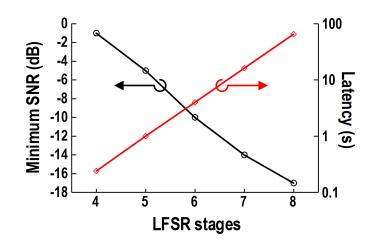


Figure 4.22: Measurement results of acoustic signature wakeup detection with various LFSR stages, showing the tradeoff between the minimum required SNR versus worst case detection latency.

clock ran at 1 kHz, and NN output data was observed every 512 ms. Once a voice activity was detected, the proposed acoustic chip sent an interrupt request to an external microcontroller via an inter-chip serial interface [94]. Then, the microcontroller sequentially waked up HP AFE chain and HP digital back-end via the serial interface. The 100-ms delay was given for AFE signal settlement before the digital back-end operation. The system clock was switched to 8 kHz, and frame length of HP NN was 16 ms (measured in 128-pt FFT and 64 features case). The acoustic system also compressed audio with a frame length of 24 ms in the HP mode. The HP detection threshold was set to achieve a high non-speech hit rate and accurate false alarm removal (97%/25% non-speech/speech hit rates, measured with a 128-80-20-2 NN model and 256-pt FFT). When there was no voice for long enough time, the acoustic system returned to the ULP mode.

## 4.7 Summary

This chapter demonstrated the design of a sub- $\mu$ W voice and non-voice acoustic activity detection chip. By using mixer-based sequential frequency scanning operation,

CLK MBUS_D MBUS_CK NNOUT COMPOUT	ULP Speech Detected	+10 s HP N	on-Speech Detected
+60 ms +70 ms +80 ms HP AFE Turn OI	פרומהם ברומיביו ערופיביו כם ברום מקומהם אם ברומהם מהפרהם המומהם הם ברום הם ברום הם ברום הם היום	s +50 ms +60 m +70 ms +80 ms +90 m HP Digital Turn ON	B HP <sup>+10 ms +20 ms +30 ms +4</sup>
	eural Network Output		16ms 24ms

Figure 4.23: Measured waveform of the acoustic system that switches between ULP and HP modes.

the feature extraction power is reduced by  $4\times$ . Table 4.1 compares the proposed feature extractor with prior works. Although [42] shows the lowest power consumption, the signal bandwidth is limited to under 500 Hz. This work achieves the lowest normalized power consumption, calculated in the same manner as in [76], which reflects the power normalized to the number of channels and signal bandwidth. Moreover, this work achieves the best front end dynamic range, thanks to the proposed amplifier design.

Table 4.2 shows the comparison of this work with prior state-of-the-art VAD systems. While this design consumes the lowest power, it is worthwhile to also consider the latency or throughput. For example, [76] has better energy efficiency in terms of classification/W/s than this work. However, it is not always possible to scale the power consumption of feature extraction to a lower power level with relaxed latency since it is an always-on block. In addition, there are still useful applications (e.g.,

	This Work	[76]	[42]	[75]	
Technology (nm)	180	180	180	90	
Feature Extraction Type	Mixed-signal	Analog to events	Digital	Analog	
Channel Number	16-48	16	4-16	16	
Frequency Range (Hz)	75-4k	100-5k	0.2 - 470	75-5k	
Power (nW)	60	380	10	6000	
Normalized Power $(nW)^1$	5	71	34	1186	
Dynamic Range (dB)	47	40	N/A	40	
Building Blocks	LNA, Mixer, LPF, DSP	LNA, BPF, FWR, IAF	DSP	LNA, BPF, FWR, LPF	

Table 4.1: Comparison of Feature Extractor

<sup>1</sup>Nomarlized power is calculated according to the equation in [5], normalized to 4kHz.

Table 4.2: Comparison of Voice Activity Detector (VAD)

	This Work	[76]	[62]	[75]	[74]
Technology (nm)	180	180	65	90	32
Acoustic Input	Analog mic. w/ gain stage	Analog mic. w/ gain stage	Assumed digitized	Analog mic. w/ gain stage	Assumed digitized
Classifier	Neural network	Neural network	Neural network	Decision tree	Energy-based
Classifier Topology Programmability	Yes	No	Yes	No	No
$Dataset^1$	LibriSpeech + NOISEX-92	AURORA4 + DEMAND	AURORA2	NOISEUS	N/A
Latency (ms)	512	10	10	<100	10
Power $(\mu W)$	0.142	1	22.3	6	300
$\frac{SP/Non-SP}{hit rate^2}$	91.5%/90% <sup>3</sup> @ 10dB SNR	84%/85%@ 10dB SNR	90%/90% <sup>4</sup> @ 7dB SNR	89%/85%@ 10dB SNR	97%@ N/A
Acoustic Testing Performed	Yes	No	No	No	No

 $^{1}$ All datasets are similar in speech quality.  $^{2}$ Tested electrically.

<sup>3</sup>Measured at ULP mode with 128pt-DCT, 32 feature channels, and 250Hz IF.  $^4$ Converted from EER in [3].

compressed speech recording after VAD) that can tolerate this latency, given the normal speech rate are 120–160 words per minute. Moreover, the digital backend design of this work offers greater flexibility to use various model topologies compared to [76], making this design a better approach for applications that are extremely power constrained yet require mapping to various target events, such as miniaturized battery-operated IoT sensor nodes.

## CHAPTER V

# Millimeter-Scale Wireless Audio Sensor Node

Bulky components of audio sensor node such as microphone, battery and antenna often impede compact system integration. This chapter demonstrates a fully functional and self-contained audio sensor node in millimeter-scale size including voice activity and acoustic object detection, continuous audio streaming with compression, non-volatile system storage, and transmission of the audio data over a 20m wireless link all when operating on millimeter-scale thin-film batteries with solar harvesting. The complete audio sensor is enabled by stacked-die 3D integration of multi-chip system with other system components.

#### 5.1 The First Generation

The complete system consists of 6 heterogeneous stacked ICs as shown in Figure 5.1: 1) The proposed audio processing IC described in Chapter II acquires and compresses audio signal, and streams out through a dedicated audio bus. 2) An 8Mb of custom embedded Flash introduced in Chapter II stores compressed audio at 656pJ/bit. 3) An RF transceiver co-designed with a 3D Roger-substrate antenna [95] communicates with a gateway up to 20m away. 4) The energy harvester charges the battery using stacked photovoltaic (PV) cells [96] and also protects it from reverse current. 5) The power management unit (PMU) [97] converts battery voltage to 1.2V

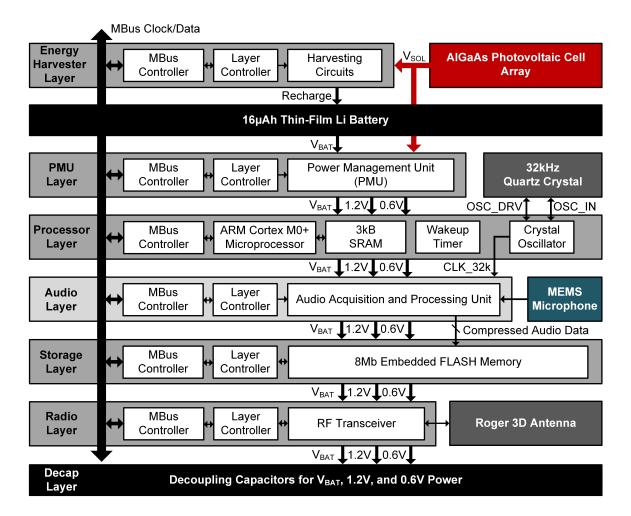
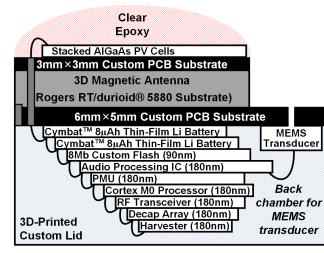


Figure 5.1: Overall block diagram of the first generation audio sensor node.

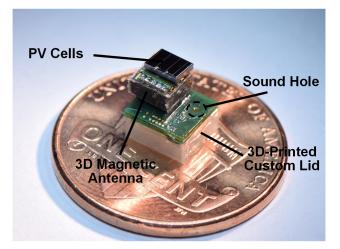
and 0.6V to provide multiple voltage domains to the ICs. 6) An ARM Cortex M0 processor coordinates system operation and enables additional signal processing such as event detection. The stacked ICs communicate via ultra-low power Mbus [98].

The system integration strategy is carefully devised to achieve minimal volume as shown in Figure 5.2. On the bottom side of a custom  $6 \times 5 \text{ mm}^2$  PCB substrate, we stack 2 rechargeable thin-film Li batteries together with ICs. We place a MEMS transducer directly adjacent to the stacked ICs to minimize the system volume and improve SNR by limiting the parasitic capacitance between the transducer and audio processing IC. A 3D-printed custom lid covers all electronics, including a 32kHz crystal and 3 capacitors for the RF transceiver to generate an acoustic back chamber. By combining the cavity for the sound chamber with the location of all electronics, system volume is aggressively reduced and also protects the electronics from light. At the same time, air volume is also increased compared to a commercial package which improves the microphone's sensitivity and low frequency response. The top side contains a sound hole for air passage, and a 3D magnetic dipole antenna. The magnetic dipole does not require physical separation from the electronics, further enabling compact integration. PV cells are mounted on top of the antenna and covered with clear epoxy to provide protection while allowing light to reach the PV cells.

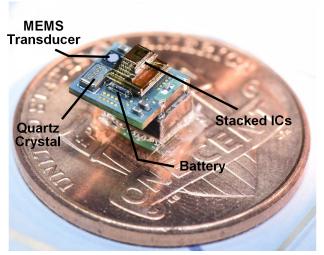
Fully functional operation, including audio acquisition, compression, storage and RF transmission, of a millimeter-scale unit identical to that pictured in Figure 5.2 was demonstrated when operating stand-alone powered only by its internal battery and energy harvesting. Measured power profile of the stand-alone operation is shown in Figure 5.3. Initially, the system is in deep sleep mode and consumes 7nW. After booting up, the processor initiates relevant chip settings and manages whole system operation by a loaded C program. After the settling time of analog circuits within a system, Flash is erased first, and then audio processing/streaming and Flash programming are performed simultaneously. When streaming is finished, whole system



(a) Cross sectional diagram



(b) Top-facing view



(c) Bottom-facing view

Figure 5.2: The complete  $6 \times 5 \times 4$  mm<sup>3</sup> audio sensor node.

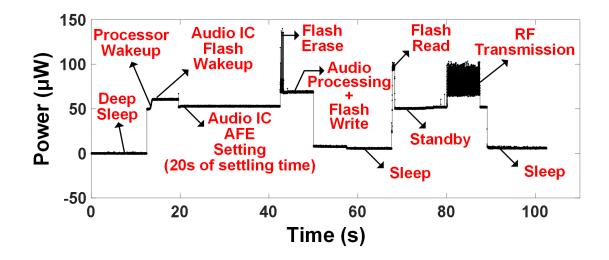


Figure 5.3: Measured power profile of audio sensor node.

goes sleep power gating mode, and the Flash retains streamed audio data during the sleep mode. When system wakes up again, the stored audio data is read by processor and delivered to transceiver chip. Finally, the audio footage is successfully transmitted to a gateway wirelessly. With harvesting from  $2.6 \times 3 \text{ mm}^2$  PV cell at 3klux,  $0.45\mu$ W of power is attained at 4V. The measured RF transmission power is  $79\mu$ W, and the system sleep power is 7.2nW. Real-time audio acquisition, compression, and streaming to Flash consumes  $68\mu$ W of power. The system storage supports ~40 mins of compressed audio streaming when  $15\times$  compression is performed (normal human speech), while the system battery supports ~1 hour of continuous audio streaming. After maximum battery usage, 6 days of charge recovery time is needed. Overall system parameters are summarized in Table 5.1.

## 5.2 The Second Generation

Although the first generation audio sensor is successfully developed to demonstrate millimeter-scale distributed audio sensing, it doesn't support low power ML capability that monitors acoustic scenes and intelligently wakes up the system. Therefore, the sensor node should be duty-cycled by using a system timer or by end-users, may

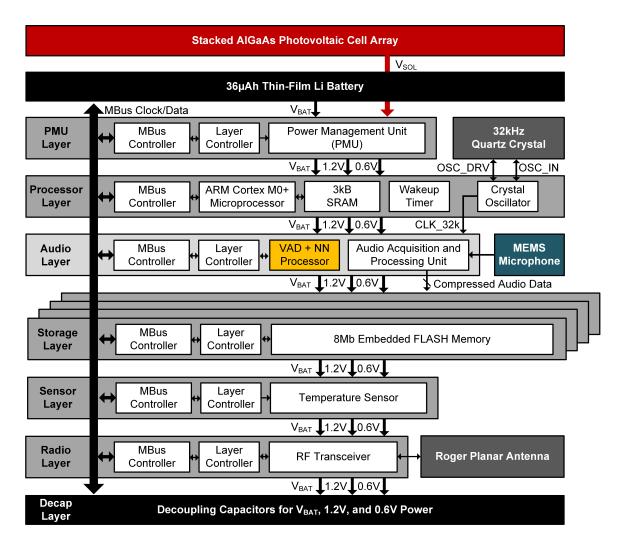


Figure 5.4: Overall block diagram of the second generation audio sensor node.

missing some events. As new techniques are proposed and developed in Chapter III and Chapter VI, we devise the second generation of millimeter-scale wireless audio sensor node in this Section.

Figure 5.4 shows a block diagram of the complete system. The acoustic signal processing chip in Chapter VI integrated with a MEMS microphone performs acoustic event/object detection by using a built-in NN processor at nW-level power. Upon an event detection, it wakes up the whole system. Depending on applications, it also acquires, compresses, and streams out audio signals. In the second generation, we increase the battery size from  $16\mu$ Ah to  $36\mu$ Ah to allow longer lifetime. Also, four

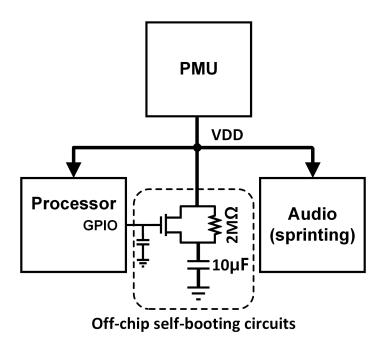


Figure 5.5: Self booting circuits using COTS components.

8Mb Flash chips (Chapter II) are stacked to provide more storage to the system (i.e. 1MB in the first generation; 4MB in the second generation). In this generation, we use 8-series-stacked PV cells [99] for solar harvesting to avoid efficiency loss from voltage up-conversion in harvester chip of the first generation, and thus to increase harvesting efficiency. In addition, we include a temperature sensor layer in the sensor node so that the system has temperature immunity. Based on the sensed temperatures, all settings (ex. analog front-end circuits, clock frequencies, and PMU current drive strength) are automatically adjusted by a processor to ensure proper operations over wide temperature range. In this system, the processor, PMU, temperature sensor, and RF transceiver are integrated in one-chip solution (CIS chip) to reduce overall sensor node volume.

In this system, the acoustic wakeup detector is always-on during the system sleep mode. In the sleep mode, the PMU output current strength is adjusted to supply a few hundreds of nA, minimizing overall system current consumption. However, although the average current consumption is nA-level, the NN processor sprints and thus instantaneously draws  $\mu$ A-level current, as explained in Chapter VI. The PMU output strength in sleep mode can not hold this surge current. Therefore, we place  $10\mu$ F of power capacitor on the board to store enough charges to maintain voltage level during the sprint operation. However, this causes another problem when the system boot-up. Initial PMU strength for booting is fixed and not enough to drive  $10\mu$ F capacitance, resulting in voltage collapse. To resolve this issue, we configure selfboot circuits shown in Figure 5.5 by using COTS power switch. Initially, the supply voltage came from PMU is weakly connected to the  $10\mu$ F power capacitor through a large resistor (2M $\Omega$ ). After system boot and the supply voltage stabilization, the processor connects the supply rail to the power capacitor using a GPIO port and a COTS transistor before the sprinting operation starts.

The system integration strategy is illustrated in Figure 5.6. On a custom  $\phi 11 \text{ mm}$ round shape Roger PCB substrate, we place 2 separate stacks: 1) main processing stack, and 2) storage stack as shown in Figure 5.7. The main processing stack consists of acoustic signal processing chip (Chapter VI), CIS chip, and 3 rechargeable thin-film Li batteries. The storage stack consists of four 8Mb embedded Flash chips (Chapter II), decoupling capacitor array chip, and PV cells. As same as in the first generation, we place a MEMS microphone directly adjacent to the audio chip to minimize the parasitic capacitance and system size. To reduce volume of sensor node, 3D-shaped antenna in the first generation is replaced with a round-shaped planar antenna. A same approach that combines the cavity for sound chamber with the location of all electronics is used in the second generation. A 3D-printed custom cylindrical wall and transparent top cover protect all electronics, including a quartz crystal and off-chip passive devices, generating acoustic back chamber at the same time. The bottom side of the board contains a sound hole for microphone air passage. All stacked dies are encapsulated by black epoxy to block the light. The PV cells are covered with clear epoxy to allow light exposure for solar harvesting and optical communication [3].

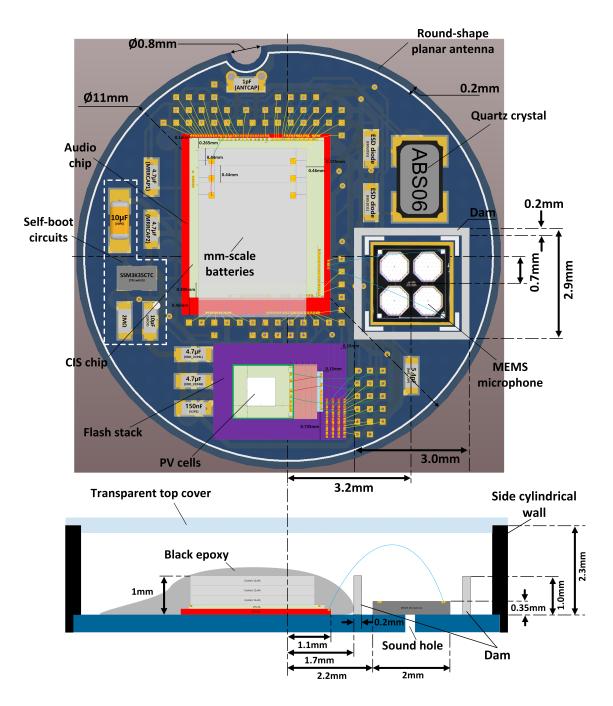


Figure 5.6: Planar and side view diagram of system integration.

**Black epoxy** 

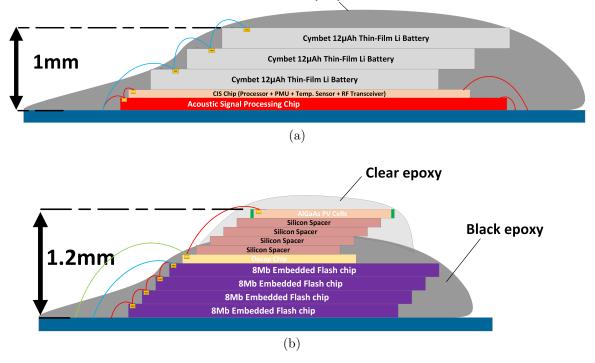


Figure 5.7: Vertical view of (a) main processing stack and (b) storage stack.

Figure 5.9 shows the measured full operation cycle of the second generation audio sensor node. After self-booting sequence and initial settings by the PMU and processor, whole sensor system goes to sleep mode. At this point, the acoustic event detector such as VAD with NN processing runs continuously as a background task. Once it detects an event of interest, the sensor node is awaken by the interrupt. The processor re-configures all the settings for the audio streaming and starts. When the HP mode NN classifier (Chapter VI) detects no-event, the processor stops audio recording and the system goes back to sleep mode. Either by end-users or timer, the system wakes up and reads the stored audio footage from Flash. Finally, the audio data is retrieved by wireless communication. The always-on acoustic detection mode consumes ~800nW including the PMU power conversion efficiency. The continuous compressed audio streaming to Flash consumes  $45\mu$ W, and wireless TX consumes  $160\mu$ W. For the acoustic event monitoring, the system sustains 7.5 days without recharge. For the continuous audio recording, the system operates 3.2 hours,

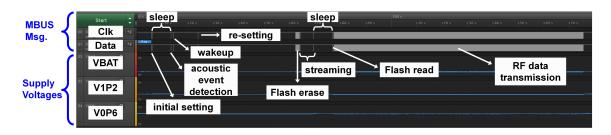


Figure 5.8: Audio sensor system operation cycle.

	1st generation	2nd generation
Dimension	6mm × $5$ mm × $4$ mm	$\phi 11 \mathrm{mm}  imes 3 \mathrm{mm}$
Storage	1MB Flash / 24kB SRAM	4MB Flash / 64kB SRAM
Processor	ARM Cortex M0	ARM Cortex M0
Battery	$16\mu Ah$ thin-film	$36\mu$ Ah thin-film
Energy harvesting	Solar, $0.45 \mu W@3klux$	Solar, $2.87\mu$ W@3klux
Audio feature	Audio streaming only	Audio streaming w/ event detection
Sleep power	$7\mathrm{nW}$	$800 nW^1$
Streaming power	$68 \mu W$	$45 \mu W$

Table 5.1: Comparison of Developed Audio Sensor Node

<sup>1</sup>Sleep with acoustic event detection

which is  $3.2 \times$  longer than the first generation. Assuming the 1% of event activity, the system streams audio data for 4.8 days without recharge if it streams only when the event occurs. In terms of system storage, ~125 mins of compressed audio recording is achieved, which is  $3.1 \times$  improvement over the first generation. The PV cells provide  $2.87\mu$ W at 4V, charging the battery within 2 days. The system parameters of the second generation are summarized in Table 5.1.

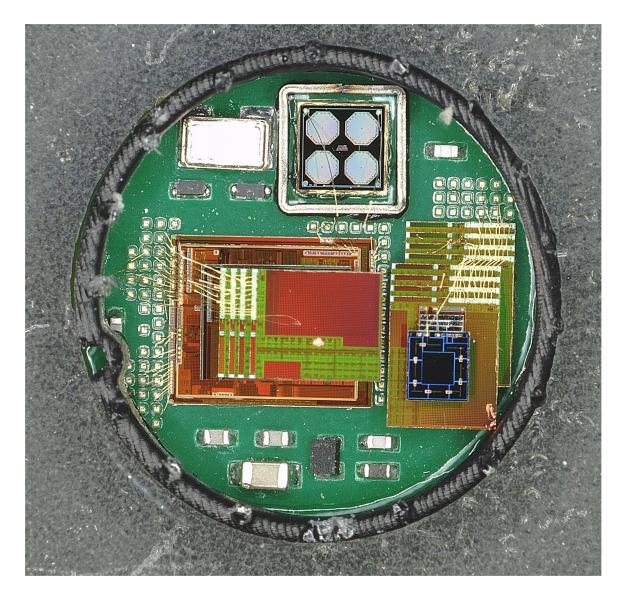


Figure 5.9: The complete  $\phi 11 \mathrm{mm}$   $\times$  3mm audio sensor node.

## CHAPTER VI

# Conclusion

Acoustic sensing modality is highly desired in extensive applications as it offers several advantages over any other sensing methods. The information carried by sound is comparable to that of image, yet requires much lower computational cost to be processed. In addition, acoustic signal can be captured omni-directionally, and also regardless of light or obstacle condition. Although it has been extensive research on millimeter-scale sensor nodes, they are often limited to simple sensing modality such as pressure or temperature. Realizing ultra-small audio sensing/processing platform that can be embedded virtually anywhere will dramatically broaden its applications such as event logging, emergency identification or object detection. The key challenge to reduce the form factor of audio sensor node is low-power operation due to the small battery size. Also, small storage size imposes a constraint on data size to be stored or processed. In this dissertation, a wide range of low-power techniques were covered starting from algorithms to a complete system level integration. Main focus has been made on reducing overall power consumption of audio sensor node, and thus making its lifetime longer while avoiding critical performance degradation. To achieve this demanding goal, all system components including algorithms, circuits, architectures, and system level designs were carefully studied and crafted.

This dissertation first presented an audio processing IC that performs audio acqui-

sition and compression, consuming  $4.7\mu W$ . A new low-power compression algorithm exploited frequency domain signal sparsity to compress raw audio samples. The proposed algorithm guaranteed constant worst-case compression ratio, but showed variable rate depending on signal sparsity. The algorithm achieved  $1000 \times$  lower complexity than CELP algorithm and  $3.9 \times$  better compression than ADPCM algorithm, maintaining similar sound quality. In this work, an efficient hardware architecture for accelerating the algorithm was also studied. Along with clock/data gating and zero skipping, the proposed architecture reduced the power consumption by 92%. The proposed compression engine showed only  $1.5\mu W$  of power consumption to provide  $4-32 \times$  real-time audio compression. Additionally, newly designed custom 8Mb embedded NOR Flash was proposed to enable seamless audio streaming by a ping-pong buffering scheme. With the ping-pong streaming, the Flash programming power was reduced by  $2.1\times$ , achieving  $38\mu W$  power. Overall, this work proved microwatt-level audio streaming is possible, realizing real-time audio acquisition on millimeter-scale energy sources.

A picowatt-level standby power neural network processor was introduced for sensor applications. In accordance with the recent success of machine learning techniques, having on-sensor inference capability is highly demanding. Due to the large computation amount of machine learning algorithms, the computing hardware often leaks high current even during the standby from tons of transistors inside. By combining custom instruction set architecture, compact SIMD microarchitecture, and ultra-low leakage SRAM memory, this dissertation work proposed a compact programmable neural network processor that can be embedded on resource-constrained sensor node. The proposed custom instruction set provided dense program to minimize required storage amount, yet showed sufficient flexibility to realize general sensor node tasks. The microarchitecture of the processor exploited parallelism for efficient processing of machine learning workloads, while still had minimal hardware complexity to reduce power consumption. Newly designed ultra-low leakage 7T SRAM memory showed 3.35 fW/bit of standby power. The proposed processor consumed only 440pW standby power, and achieved 400-GOPS/W of active energy efficiency. Among the prior works of on- or near-sensor neural network processor, this work showed the highest energy efficiency and the lowest standby power at the same time. The proposed neural network processor is integrated in an acoustic object detection sensor system, and successfully demonstrated >90% of positive detection and <3% of false alarm for 5 acoustic targets detection.

As a wakeup method for the sensor nodes, a voice and acoustic activity detector that uses a mixer-based architecture and ultra-low power neural network based classifier was proposed. By sequentially scanning 4 kHz of frequency bands and downconverting to below 500 Hz, feature extraction power consumption was reduced by  $4\times$ . The neural network processor realized computational sprinting to achieve  $12\times$  power reduction. The system also demonstrated inaudible acoustic signature detection for intentional remote silent wakeup by users while re-using a subset of the same circuit components. The measurement results of voice activity detection showed 91.5%/90%of speech/non-speech hit rates at 10 dB SNR with babble noise and 142 nW power consumption. Acoustic signature detection consumed 66 nW, successfully detecting a signature 10 dB below the noise level.

Finally, this work developed two generations of complete wireless audio sensor node with millimeter-scale form factor. This was enabled by the proposed audio processing ICs and neural network processor, integrated with a MEMS microphone, general-purpose microprocessor, 8Mb Flashes, RF transceiver with custom antenna, PV cells for energy harvesting and optical communication, and millimeter size batteries. The complete stand-alone systems achieved 1 hour (1st gen.) and 3.2 hours (2nd gen.) of speech recording and energy-autonomous operation in bright room light.

Demand on lower power, smaller form factor and longer life time for wireless

audio sensing devices will continue to increase as time passes. All of the works presented in this dissertation give subtle guidance for designing ultra-small audio devices in somewhat extreme design space. Such design techniques and approaches could possibly expedite the development of future seamless audio sensing all around.

# BIBLIOGRAPHY

## BIBLIOGRAPHY

- J. M. Kahn, R. H. Katz, and K. S. J. Pister, "Emerging challenges: Mobile networking for "smart dust"," *Journal of Communications and Networks*, vol. 2, no. 3, pp. 188–196, 2000.
- [2] B. Warneke, M. Last, B. Liebowitz, and K. S. J. Pister, "Smart dust: communicating with a cubic-millimeter computer," *Computer*, vol. 34, no. 1, pp. 44–51, 2001.
- [3] Y. Lee, G. Kim, S. Bang, Y. Kim, I. Lee, P. Dutta, D. Sylvester, and D. Blaauw, "A modular 1mm<sup>3</sup> die-stacked sensing platform with optical communication and multi-modal energy harvesting," in 2012 IEEE International Solid-State Circuits Conference, pp. 402–404, 2012.
- [4] X. Wu, I. Lee, Q. Dong, K. Yang, D. Kim, J. Wang, Y. Peng, Y. Zhang, M. Saliganc, M. Yasuda, K. Kumeno, F. Ohno, S. Miyoshi, M. Kawaminami, D. Sylvester, and D. Blaauw, "A 0.04mm 316nw wireless and batteryless sensor system with integrated cortex-m0+ processor and optical communication for cellular temperature measurement," in 2018 IEEE Symposium on VLSI Circuits, pp. 191–192, 2018.
- [5] G. Chen, M. Fojtik, D. Kim, D. Fick, J. Park, M. Seok, M. Chen, Z. Foo, D. Sylvester, and D. Blaauw, "Millimeter-scale nearly perpetual sensor system with stacked battery and solar cells," in 2010 IEEE International Solid-State Circuits Conference - (ISSCC), pp. 288–289, 2010.
- [6] H. Kim, G. Kim, Y. Lee, Z. Foo, D. Sylvester, D. Blaauw, and D. Wentzloff, "A 10.6mm<sup>3</sup> fully-integrated, wireless sensor node with 8ghz uwb transmitter," in 2015 Symposium on VLSI Circuits (VLSI Circuits), pp. C202–C203, 2015.
- [7] T. Kang, I. Lee, S. Oh, T. Jang, Y. Kim, H. Ahn, G. Kim, S. Shin, S. Jeong, D. Sylvester, and D. Blaauw, "A 1.7×4.1×2 mm<sup>3</sup> fully integrated ph sensor for implantable applications using differential sensing and drift-compensation," in 2019 Symposium on VLSI Circuits, pp. C310–C311, 2019.
- [8] G. Kim, Y. Lee, Zhiyoong Foo, P. Pannuto, Ye-Sheng Kuo, B. Kempke, M. H. Ghaed, Suyoung Bang, Inhee Lee, Yejoong Kim, Seokhyeon Jeong, P. Dutta, D. Sylvester, and D. Blaauw, "A millimeter-scale wireless imaging system with

continuous motion detection and energy harvesting," in 2014 Symposium on VLSI Circuits Digest of Technical Papers, pp. 1–2, 2014.

- [9] Z. C. Taysi, M. A. Guvensan, and T. Melodia, "Tinyears: Spying on house appliances with audio sensor nodes," in *Proceedings of the 2nd ACM Workshop* on Embedded Sensing Systems for Energy-Efficiency in Building, BuildSys '10, (New York, NY, USA), p. 31–36, Association for Computing Machinery, 2010.
- [10] G. Dekkers, S. Lauwereins, B. Thoen, M. W. Adhana, H. Brouckxon, B. Van den Bergh, T. van Waterschoot, B. Vanrumste, M. Verhelst, and P. Karsmakers, "The sins database for detection of daily activities in a home environment using an acoustic sensor network," 2017.
- [11] Z. Jiang, Z. Zhang, and A. Maxwell, "Extraction of structural modal information using acoustic sensor measurements and machine learning," *Journal of Sound* and Vibration, vol. 450, pp. 156 – 174, 2019.
- [12] Z. Sheng, S. Pfersich, A. Eldridge, J. Zhou, D. Tian, and V. C. M. Leung, "Wireless acoustic sensor networks and edge computing for rapid acoustic monitoring," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 1, pp. 64–74, 2019.
- [13] F. Sánchez-Rosario, D. Sánchez-Rodríguez, J. B. Alonso-Hernández, C. M. Travieso-González, I. Alonso-González, C. Ley-Bosch, C. Ramírez-Casañas, and M. A. Quintana-Suárez, "A low consumption real time environmental monitoring system for smart cities based on zigbee wireless sensor network," in 2015 International Wireless Communications and Mobile Computing Conference (IWCMC), pp. 702–707, 2015.
- [14] M. Rach, H. Gomis, O. Granado, M. Malumbres, A. Campoy, and J. Martín, "On the design of a bioacoustic sensor for the early detection of the red palm weevil," *Sensors*, vol. 13, p. 1706–1729, Jan 2013.
- [15] A. F. Smeaton and M. McHugh, "Towards event detection in an audio-based sensor network," in *Proceedings of the Third ACM International Workshop on Video Surveillance and Sensor Networks*, VSSN '05, (New York, NY, USA), p. 87–94, Association for Computing Machinery, 2005.
- [16] G. Zhao, H. Ma, Y. Sun, H. Luo, and X. Mao, "Enhanced surveillance platform with low-power wireless audio sensor networks," in *Proceedings of the 2011 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, WOWMOM '11, (USA), p. 1–9, IEEE Computer Society, 2011.
- [17] A. Griffin, A. Alexandridis, D. Pavlidi, Y. Mastorakis, and A. Mouchtaris, "Localizing multiple audio sources in a wireless acoustic sensor network," *Signal Process.*, vol. 107, p. 54–67, Feb. 2015.
- [18] X. Han and M. A. Rashid, "Gesture and voice control of internet of things," in 2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA), pp. 1791–1795, 2016.

- [19] M. Berglund, J. Nelson, and D. Picovici, "Voice user interface for understanding wireless sensor networks," in 2006 IET Irish Signals and Systems Conference, pp. 173–177, 2006.
- [20] J. R. Agre, L. P. Clare, G. J. Pottie, and N. P. Romanov, "Development platform for self-organizing wireless sensor networks," in *Unattended Ground Sensor Technologies and Applications* (E. M. Carapezza, D. B. Law, and K. T. Stalker, eds.), vol. 3713, pp. 257 – 268, International Society for Optics and Photonics, SPIE, 1999.
- [21] R. Min, M. Bhardwaj, Seong-Hwan Cho, A. Sinha, E. Shih, A. Wang, and A. Chandrakasan, "An architecture for a power-aware distributed microsensor node," in 2000 IEEE Workshop on SiGNAL PROCESSING SYSTEMS. SiPS 2000. Design and Implementation (Cat. No.00TH8528), pp. 581–590, 2000.
- [22] J. Beutel, O. Kasten, and M. Ringwald, "Btnodes a distributed platform for sensor nodes," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, SenSys '03, (New York, NY, USA), p. 292–293, Association for Computing Machinery, 2003.
- [23] A. Savvides and M. B. Srivastava, "A distributed computation platform for wireless embedded sensing," in *Proceedings. IEEE International Conference on Computer Design: VLSI in Computers and Processors*, pp. 220–225, 2002.
- [24] L. Nachman, R. Kling, R. Adler, J. Huang, and V. Hummel, "The intel mote platform: a bluetooth-based sensor network for industrial monitoring," in *IPSN* 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005., pp. 437–442, 2005.
- [25] J. L. Hill and D. E. Culler, "Mica: a wireless platform for deeply embedded networks," *IEEE Micro*, vol. 22, no. 6, pp. 12–24, 2002.
- [26] J. Polastre, R. Szewczyk, and D. Culler, "Telos: enabling ultra-low power wireless research," in *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks*, 2005., pp. 364–369, 2005.
- [27] Libelium, Waspmote. http://www.libelium.com/products/waspmote/, 2008.
- [28] MEMSIC, LOTUS. http://www.memsic.com/userfiles/files/DataSheets/ WSN/6020-0705-01\_A\_LOTUS.pdf, 2011.
- [29] The Samraksh Company, .NOW with eMote. https://samraksh.com/ products/small-battery-powered-computers/31-product-pages/ product-small-battery-powered-computers/60-emote-now, 2012.
- [30] V. Berisha, Homin Kwon, and A. Spanias, "Real-time acoustic monitoring using wireless sensor motes," in 2006 IEEE International Symposium on Circuits and Systems, pp. 4 pp.–850, 2006.

- [31] G. Wichern, H. Kwon, A. Spanias, A. Fink, and H. Thornburg, "Continuous observation and archival of acoustic scenes using wireless sensor networks," in 2009 16th International Conference on Digital Signal Processing, pp. 1–6, 2009.
- [32] Crossbow Technology Inc., MICAz, 2004.
- [33] M. Rach, H. Gomis, O. Granado, M. Malumbres, A. Campoy, and J. Martín, "On the design of a bioacoustic sensor for the early detection of the red palm weevil," *Sensors*, vol. 13, p. 1706–1729, Jan 2013.
- [34] M. A. Guvensan, Z. C. Taysi, and T. Melodia, "Energy monitoring in residential spaces with audio sensor nodes: Tinyears," Ad Hoc Networks, vol. 11, no. 5, pp. 1539 – 1555, 2013.
- [35] L. Nachman, J. Huang, J. Shahabdeen, R. Adler, and R. Kling, "Imote2: Serious computation at the edge," in 2008 International Wireless Communications and Mobile Computing Conference, pp. 1118–1123, 2008.
- [36] M. Magno, D. Brunelli, L. Sigrist, R. Andri, L. Cavigelli, A. Gomez, and L. Benini, "Infinitime: Multi-sensor wearable bracelet with human body harvesting," *Sustainable Computing: Informatics and Systems*, vol. 11, pp. 38 – 49, 2016. SI: IGCC 2014.
- [37] V. Etacheri, R. Marom, R. Elazari, G. Salitra, and D. Aurbach, "Challenges in the development of advanced li-ion batteries: a review," *Energy Environ. Sci.*, vol. 4, pp. 3243–3262, 2011.
- [38] A. Bilbao, D. Hoover, J. Rice, and J. Chapman, "Ultra-low power wireless sensing for long-term structural health monitoring," in *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2011* (M. Tomizuka, ed.), vol. 7981, pp. 87 – 100, International Society for Optics and Photonics, SPIE, 2011.
- [39] G. Zhao, H. Ma, Y. Sun, and H. Luo, "Design and implementation of enhanced surveillance platform with low-power wireless audio sensor network," *International Journal of Distributed Sensor Networks*, vol. 8, no. 5, p. 854325, 2012.
- [40] C. Pham, P. Cousin, and A. Carer, "Real-time on-demand multi-hop audio streaming with low-resource sensor motes," in 39th Annual IEEE Conference on Local Computer Networks Workshops, pp. 539–543, 2014.
- [41] Q. Dong, Y. Kim, I. Lee, M. Choi, Z. Li, J. Wang, K. Yang, Y. Chen, J. Dong, M. Cho, G. Kim, W. Chang, Y. Chen, Y. Chih, D. Blaauw, and D. Sylvester, "A 1mb embedded nor flash memory with 39μw program power for mm-scale high-temperature sensor nodes," in 2017 IEEE International Solid-State Circuits Conference (ISSCC), pp. 198–199, 2017.

- [42] S. Jeong, Y. Chen, T. Jang, J. M. Tsai, D. Blaauw, H. Kim, and D. Sylvester, "Always-on 12-nw acoustic sensing and object recognition microsystem for unattended ground sensor nodes," *IEEE Journal of Solid-State Circuits*, vol. 53, no. 1, pp. 261–274, 2018.
- [43] J. Rothweiler, "Polyphase quadrature filters-a new subband coding technique," in ICASSP '83. IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 8, pp. 1280–1283, 1983.
- [44] K. Konstantinides, "Fast subband filtering in mpeg audio coding," IEEE Signal Processing Letters, vol. 1, no. 2, pp. 26–28, 1994.
- [45] H. Mitani, K. Matsubara, H. Yoshida, T. Hashimoto, H. Yamakoshi, S. Abe, T. Kono, Y. Taito, T. Ito, T. Krafuji, K. Noguchi, H. Hidaka, and T. Yamauchi, "7.6 a 90nm embedded 1t-monos flash macro for automotive applications with 0.07mj/8kb rewrite energy and endurance over 100m cycles under tj of 175°c," in 2016 IEEE International Solid-State Circuits Conference (ISSCC), pp. 140–141, 2016.
- [46] S. Jeloka, J. Lee, Z. Li, J. Shah, Q. Dong, K. Yang, D. Sylvester, and D. Blaauw, "An ultra-wide program, 122pj/bit flash memory using charge recycling," in 2017 Symposium on VLSI Circuits, pp. C196–C197, 2017.
- [47] M. Cho, S. Oh, S. Jeong, Y. Zhang, I. Lee, Y. Kim, L. Chuo, D. Kim, Q. Dong, Y. Chen, M. Lim, M. Daneman, D. Blaauw, D. Sylvester, and H. Kim, "A 6×5×4mm<sup>3</sup> general purpose audio sensor node with a 4.7µw audio processing ic," in 2017 Symposium on VLSI Circuits, pp. C312–C313, 2017.
- [48] G. Serpen, J. Li, and L. Liu, "AI-WSN: Adaptive and intelligent wireless sensor network," *Procedia Computer Science*, vol. 20, pp. 406 – 413, 2013.
- [49] I. Lobachev, R. Maleryk, S. Antoschuk, D. Filiahin, and M. Lobachev, "Integration of neural networks into smart sensor networks," in *IEEE Int. Conf. on Dependable Systems, Services and Technologies (DESSERT)*, pp. 544–548, 2018.
- [50] B. Liu, H. Cai, Z. Wang, Y. Sun, Z. Shen, W. Zhu, Y. Li, Y. Gong, W. Ge, J. Yang, and L. Shi, "A 22nm, 10.8μw/15.1μw dual computing modes high power-performance-area efficiency domained background noise aware keywordspotting processor," *IEEE Transactions on Circuits and Systems I: Regular Papers*, pp. 1–14, 2020.
- [51] M. Villemur, P. Julian, T. Figliolia, and A. G. Andreou, "7 tops/w cellular neural network processor core for intelligent internet-of-things," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 7, pp. 1324–1328, 2020.
- [52] S. Chang, B. Wu, Y. Liou, R. Zheng, P. Lee, T. Chiueh, and T. Liu, "An ultralow-power dual-mode automatic sleep staging processor using neural-networkbased decision tree," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 9, pp. 3504–3516, 2019.

- [53] S. Zheng, P. Ouyang, D. Song, X. Li, L. Liu, S. Wei, and S. Yin, "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*, vol. 66, no. 12, pp. 4648–4661, 2019.
- [54] F. Iandola and K. Keutzer, "Keynote: Small neural nets are beautiful: Enabling embedded systems with small deep-neural- network architectures," in *IEEE/ACM Int. Conf. on Hardware/Software Codesign and System Synthesis* (CODES+ISSS), pp. 1–10, 2017.
- [55] D. Valencia, S. F. Fard, and A. Alimohammad, "An artificial neural network processor with a custom instruction set architecture for embedded applications," *IEEE Transactions on Circuits and Systems I: Regular Papers*, pp. 1–11, 2020.
- [56] J. Rust and S. Paul, "Design and implementation of a neurocomputing asip for environmental monitoring in wsn," in 2012 19th IEEE International Conference on Electronics, Circuits, and Systems (ICECS 2012), pp. 129–132, 2012.
- [57] P. N. Whatmough, S. K. Lee, D. Brooks, and G. Wei, "Dnn engine: A 28-nm timing-error tolerant sparse deep neural network processor for iot applications," *IEEE Journal of Solid-State Circuits*, vol. 53, no. 9, pp. 2722–2731, 2018.
- [58] S. Oh, M. Cho, Z. Shi, J. Lim, Y. Kim, S. Jeong, Y. Chen, R. Rothe, D. Blaauw, H. Kim, and D. Sylvester, "An acoustic signal processing chip with 142-nw voice activity detection using mixer-based sequential frequency scanning and neural network classification," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 11, pp. 3005–3016, 2019.
- [59] J. Cong, M. A. Ghodrat, M. Gill, B. Grigorian, K. Gururaj, and G. Reinman, "Accelerator-rich architectures: Opportunities and progresses," in 2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC), pp. 1–6, 2014.
- [60] S. Liu, Z. Du, J. Tao, D. Han, T. Luo, Y. Xie, Y. Chen, and T. Chen, "Cambricon: An instruction set architecture for neural networks," in 2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA), pp. 393–405, 2016.
- [61] S. Bang, J. Wang, Z. Li, C. Gao, Y. Kim, Q. Dong, Y. Chen, L. Fick, X. Sun, R. Dreslinski, T. Mudge, H. S. Kim, D. Blaauw, and D. Sylvester, "A 288μw programmable deep-learning processor with 270kb on-chip weight storage using non-uniform memory hierarchy for mobile intelligence," in *IEEE International Solid-State Circuits Conference (ISSCC)*, pp. 250–251, 2017.
- [62] M. Price, J. Glass, and A. P. Chandrakasan, "A low-power speech recognizer and voice activity detector using deep neural networks," *IEEE Journal of Solid-State Circuits*, vol. 53, no. 1, pp. 66–75, 2018.

- [63] B. Reagen, R. Adolf, and P. Whatmough, Deep Learning for Computer Architects. Morgan & Claypool, 2017.
- [64] D. Rakanovic and R. Struharik, "Implementation of application specific instruction-set processor for the artificial neural network acceleration using lisa adl," in 2017 IEEE East-West Design Test Symposium (EWDTS), pp. 1–6, 2017.
- [65] B. Moons, K. Goetschalckx, N. Van Berckelaer, and M. Verhelst, "Minimum energy quantized neural networks," in 2017 51st Asilomar Conference on Signals, Systems, and Computers, pp. 1921–1925, 2017.
- [66] C. Sakr, Y. Kim, and N. Shanbhag, "Analytical guarantees on numerical precision of deep neural networks," vol. 70 of *Proceedings of Machine Learning Research*, (International Convention Centre, Sydney, Australia), pp. 3007–3016, PMLR, 06–11 Aug 2017.
- [67] D. Shin, J. Lee, J. Lee, J. Lee, and H. Yoo, "Dnpu: An energy-efficient deeplearning processor with heterogeneous multi-core architecture," *IEEE Micro*, vol. 38, no. 5, pp. 85–93, 2018.
- [68] L. Chang, D. Fried, J. Hergenrother, J. W. Sleight, R. Dennard, R. K. Montoye, L. Sekaric, S. mcnab, A. Topol, C. D. Adams, K. W. Guarini, and W. Haensch, "Stable sram cell design for the 32 nm node and beyond," *Digest of Technical Papers. Symposium on VLSI Technology*, pp. 128–129, 2005.
- [69] J. Kulkarni, B. Geuskens, T. Karnik, M. Khellah, J. Tschanz, and V. De, "Capacitive-coupling wordline boosting with self-induced vcc collapse for write vmin reduction in 22-nm 8t sram," in 2012 IEEE International Solid-State Circuits Conference, pp. 234–236, 2012.
- [70] L. Chang, R. K. Montoye, Y. Nakamura, K. A. Batson, R. J. Eickemeyer, R. H. Dennard, W. Haensch, and D. Jamsek, "An 8t-sram for variability tolerance and low-voltage operation in high-performance caches," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 4, pp. 956–963, 2008.
- [71] G. Chen, M. Fojtik, D. Kim, D. Fick, J. Park, M. Seok, M. Chen, Z. Foo, D. Sylvester, and D. Blaauw, "Millimeter-scale nearly perpetual sensor system with stacked battery and solar cells," in 2010 IEEE International Solid-State Circuits Conference - (ISSCC), pp. 288–289, 2010.
- [72] Y. Kim, Robust Circuit Design for Low-Voltage VLSI. University of Michigan, 2015.
- [73] J. S. P. Giraldo and M. Verhelst, "Laika: A 5uw programmable lstm accelerator for always-on keyword spotting in 65nm cmos," in ESSCIRC 2018 - IEEE 44th European Solid State Circuits Conference (ESSCIRC), pp. 166–169, 2018.

- [74] A. Raychowdhury, C. Tokunaga, W. Beltman, M. Deisher, J. W. Tschanz, and V. De, "A 2.3 nj/frame voice activity detector-based audio front-end for contextaware system-on-chip applications in 32-nm cmos," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 8, pp. 1963–1969, 2013.
- [75] K. Badami, S. Lauwereins, W. Meert, and M. Verhelst, "Context-aware hierarchical information-sensing in a 6μw 90nm cmos voice activity detector," in 2015 IEEE International Solid-State Circuits Conference - (ISSCC) Digest of Technical Papers, pp. 1–3, 2015.
- [76] M. Yang, C. Yeh, Y. Zhou, J. P. Cerqueira, A. A. Lazar, and M. Seok, "Design of an always-on deep neural network-based 1- μ w voice activity detector aided with a customized software model for analog feature extraction," *IEEE Journal* of Solid-State Circuits, vol. 54, no. 6, pp. 1764–1777, 2019.
- [77] M. Cho, S. Oh, Z. Shi, J. Lim, Y. Kim, S. Jeong, Y. Chen, D. Blaauw, H. Kim, and D. Sylvester, "A 142nw voice and acoustic activity detection chip for mmscale sensor nodes using time-interleaved mixer-based frequency scanning," in 2019 IEEE International Solid- State Circuits Conference - (ISSCC), pp. 278– 280, 2019.
- [78] Y. Chen, M. Cho, S. Jeong, D. Blaauw, D. Sylvester, and H. Kim, "A dual-stage, ultra-low-power acoustic event detection system," in 2016 IEEE International Workshop on Signal Processing Systems (SiPS), pp. 213–218, 2016.
- [79] Jongseo Sohn, Nam Soo Kim, and Wonyong Sung, "A statistical model-based voice activity detection," *IEEE Signal Processing Letters*, vol. 6, no. 1, pp. 1–3, 1999.
- [80] N. Ryant, M. Liberman, and J. Yuan, "Speech activity detection on youtube using deep neural networks," in *INTERSPEECH*, 2013.
- [81] D. Ying, Y. Yan, J. Dang, and F. K. Soong, "Voice activity detection based on an unsupervised learning framework," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 8, pp. 2624–2633, 2011.
- [82] I. Tashev and S. Mirsamadi, "Dnn-based causal voice activity detector," in Information Theory and Applications Workshop, 02 2016.
- [83] G. Ferroni, R. Bonfigli, E. Principi, S. Squartini, and F. Piazza, "A deep neural network approach for voice activity detection in multi-room domestic scenarios," in 2015 International Joint Conference on Neural Networks (IJCNN), pp. 1–8, 2015.
- [84] S. M. R. Nahar and A. Kai, "Robust voice activity detector by combining sequentially trained deep neural networks," in 2016 International Conference On Advanced Informatics: Concepts, Theory And Application (ICAICTA), pp. 1–5, 2016.

- [85] J. Kim and M. Hahn, "Voice activity detection using an adaptive context attention model," *IEEE Signal Processing Letters*, vol. 25, no. 8, pp. 1181–1185, 2018.
- [86] Z. Fan, Z. Bai, X. Zhang, S. Rahardja, and J. Chen, "Auc optimization for deep learning based voice activity detection," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6760–6764, 2019.
- [87] X. Zhang and J. Wu, "Deep belief networks based voice activity detection," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 4, pp. 697–710, 2013.
- [88] X.-L. Zhang and D. Wang, "Boosted deep neural networks and multi-resolution cochleagram features for voice activity detection," in *INTERSPEECH*, 2014.
- [89] S. Mun, S. Shon, W. Kim, D. K. Han, and H. Ko, "Deep neural network based learning and transferring mid-level audio features for acoustic scene classification," in 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 796–800, 2017.
- [90] S. Thomas, G. Saon, M. V. Segbroeck, and S. S. Narayanan, "Improvements to the ibm speech activity detection system for the darpa rats program," in 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4500–4504, 2015.
- [91] S. Oh, T. Jang, K. D. Choo, D. Blaauw, and D. Sylvester, "A 4.7µw switchedbias mems microphone preamplifier for ultra-low-power voice interfaces," in 2017 Symposium on VLSI Circuits, pp. C314–C315, 2017.
- [92] P. Harpe, H. Gao, R. v. Dommele, E. Cantatore, and A. H. M. van Roermund, "A 0.20 mm<sup>2</sup> 3 nw signal acquisition ic for miniature sensor nodes in 65 nm cmos," *IEEE Journal of Solid-State Circuits*, vol. 51, no. 1, pp. 240–248, 2016.
- [93] F. Beritelli, S. Casale, and A. Cavallaero, "A robust voice activity detector for wireless communications using soft computing," *IEEE Journal on Selected Areas* in Communications, vol. 16, no. 9, pp. 1818–1829, 1998.
- [94] P. Pannuto, Y. Lee, Y. Kuo, Z. Foo, B. Kempke, G. Kim, R. G. Dreslinski, D. Blaauw, and P. Dutta, "Mbus: A system integration bus for the modular microscale computing class," *IEEE Micro*, vol. 36, no. 3, pp. 60–70, 2016.
- [95] L. Chuo, Y. Shi, Z. Luo, N. Chiotellis, Z. Foo, G. Kim, Y. Kim, A. Grbic, D. Wentzloff, H. Kim, and D. Blaauw, "A 915mhz asymmetric radio using qenhanced amplifier for a fully integrated 3×3×3mm3 wireless sensor node with 20m non-line-of-sight communication," in 2017 IEEE International Solid-State Circuits Conference (ISSCC), pp. 132–133, 2017.

- [96] I. Lee, W. Lim, A. Teran, J. Phillips, D. Sylvester, and D. Blaauw, "A >78%efficient light harvester over 100-to-100klux with reconfigurable pv-cell network and mppt circuit," in 2016 IEEE International Solid-State Circuits Conference (ISSCC), pp. 370–371, 2016.
- [97] W. Jung, J. Gu, P. D. Myers, M. Shim, S. Jeong, K. Yang, M. Choi, Z. Foo, S. Bang, S. Oh, D. Sylvester, and D. Blaauw, "A 60%-efficiency 20nw-500μw trioutput fully integrated power management unit with environmental adaptation and load-proportional biasing for iot systems," in 2016 IEEE International Solid-State Circuits Conference (ISSCC), pp. 154–155, 2016.
- [98] P. Pannuto, Y. Lee, Y. Kuo, Z. Foo, B. Kempke, G. Kim, R. G. Dreslinski, D. Blaauw, and P. Dutta, "Mbus: An ultra-low power interconnect bus for next generation nanopower systems," in 2015 ACM/IEEE 42nd Annual International Symposium on Computer Architecture (ISCA), pp. 629–641, 2015.
- [99] E. Moon, I. Lee, D. Blaauw, and J. D. Phillips, "High-efficiency photovoltaic modules on a chip for millimeter-scale energy harvesting," *Progress in Photo*voltaics: Research and Applications, vol. 27, no. 6, pp. 540–546, 2019.