Adaptive Acoustic Beamforming and Speech Processing Front-end

by

Taewook Kang

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
2021

Doctoral Committee:

Professor Michael P. Flynn, Chair
Professor Karl Grosh
Professor David D. Wentzloff
Associate Professor Zhengya Zhang

Taewook Kang

twkang@umich.edu

ORCID iD: 0000-0002-5972-4789

Acknowledgments

It was an incredible five years of journey at the University of Michigan. First, I would like to thank my parents, brother, and all other families for their unconditional love and support.

Next, I thank Michael Flynn. He is primarily my research advisor, but he is also a life mentor, big brother, and friend. It is always such a joy to talk with him (maybe not always, and that is the way a graduate school is), and he gives me so many great ideas and advice. He is truly supportive whenever I need help. I have already told him multiple times, but he is an amazing person and the best advisor.

I would like to thank the committees, Professor Karl Grosh, David Wentzloff, and Zhengya Zhang, for their valuable advice and time spent on my thesis. I thank David Blaauw for his advising. I also would like to thank all the staff at the University of Michigan for their support.

I acknowledge Mohammad Haghighat and Intel for their funding and research advice.

I thank my labmates in Flynn's research group: John Bell, Matthew Belz, Peter Brown, Hsiang-Wen Chen, Justin Correll, Linda Gong, Lu Jie, Seungjong Lee, Rundao Lu, Seungheun Song, Christine Weston, Boyi Zheng, Faustine, and other new students. It is such a pleasure to hang out and work with them. Also, I thank all of my friends and the Life Science Orchestra community. I am fortunate to meet those amazing people.

I will miss this beautiful city, Ann Arbor.

Table of Contents

List of Tables

List of Abbreviations

CMOS	complementary metal-oxide-semiconductor
CT	continuous-time
DAC	digital-to-analog converter
DAS	delay-and-sum
DASBF	delay-and-sum beamforming
DNN	deep neural networks
DOA	direction of arrival
DSM	delta-sigma modulator
DSP	digital signal processor
DT	discrete-time
DTDAS	DOA tracking DAS beamformer
DWA	data weighted averaging
ELDC	excess-loop-delay compensation
ENOB	effective number of bits
FE	feature extractor
FFT	fast Fourier transform
FIR	finite impluse response
GABF	greedy ABF
GBM	greedy blocking matrix
GMB	greedy blocking matrix
GSC	generalized sidelobe canceller
IFFT	inverse FFT
IIR	infinite impulse response
KWS	keyword spotting

LCMV	linear constraint minimum variance
LP	low power
MAC	multiplier–accumulator
MC	multiple-input-canceller
MVDR	minimum variance distortionless response
NCAF	norm-constrained adaptive filter
NLMS	normalized-least-mean-squares
NSSAR	noise-shaping SAR
NTF	noise transfer function
OSR	oversampling ratio
PVT	process, voltage, and temperature
RGSC	robust generalized sidelobe canceller
SAR	successive-approximation-register
SDM	sigma-delta modeulator (or DSM)
SINR	Signal-to-interference-and-noise ratio
SNDR	signal-to-noise-and-distortion ratio
SNR	signal-to-noise ratio
SPI	serial peripheral interface
UCA	uniform circular array
ULA	uniform linear array
VAD	voice activity detector

List of Figures

Abstract

Noise is a primary factor limiting the accuracy of automatic speech recognition (ASR). Multi-channel beamforming is essential to suppress noise and enhance the desired speech signal.

This thesis presents three fully-integrated ASR frontend systems that suppress noise and increase speech recognition accuracy in a noisy environment. The thesis focuses on the ASR frontend, which includes ADCs, adaptive beamforming, and feature extraction. We take advantage of the bitstream output of sigma-delta modulation (SDM) for fine delay resolution. We present three different beamformer prototypes with power/area-efficient hardware implementations.

The first system (Chapter 2) makes use of the synergy between data conversion and signal processing. It combines eight-channel delay-and-sum beamforming with frequency-selective beamforming and a 60-feature Mel frequency extractor to enable constant-directivity beamforming. The system improves the angular resolution of beamsteering by directly processing the raw bitstream outputs of third-order SDMs. The 40nm CMOS prototype has an active area of 1.1mm^2 and consumes 4mW. It improves the keyword spotting (KWS) accuracy from 73% to 93% using a DNN trained with noiseless speech.

The system in Chapter 3 combines a four-channel adaptive beamformer and a 40 feature Mel frequency extractor. The prototype processes the bitstream output of a 3rd order delta-sigma modulator output for accurate steering. For a given steering vector, the beamformer adaptively places a null in the noise direction by using a robust generalized sidelobe canceller (RGSC). Hardware sharing and DSP clock optimization reduce area and power consumption. It is fabricated in 40nm CMOS, occupies an active area of 0.89mm^2, and consumes 0.65mW. The prototype

beamformer improves speech recognition accuracy in noisy conditions from 64% to 90% using DNN trained with noisy speech.

Finally, the third system in Chapter 4 presents a four-channel greedy adaptive beamformer and a multi-mode ADC. The proposed system adapts beamforming and ADC performance to optimize power consumption depending on the target signal and noise. The multi-mode ADC can operate as a continuous-time noise-shaping SAR ADC (CT NSSAR) (80dBA/12μW), NSSAR ADC (65dBA/5.8μW), or as a SAR ADC (40dBA/1.5μW). The direct output of CTNSSAR enables the newly proposed greedy adaptive beamformer, which can track the direction of arrival (DOA) of the target signal, reduce signal distortion and power consumption. The 40nm CMOS prototype occupies 0.93mm^2 and consumes 157μW in high-performance mode. It improves KWS accuracy from 54% to 83% in the presence of spoken-word interference using a DNN trained with noisy speech.

Chapter 1. Introduction

1.1. Background

Deep Neural Networks (DNN) bring significant improvements in the performance of the Automatic Speech Recognition (ASR) systems [1]. Moreover, the demand for ASR keeps increasing. The global market for smart speakers is expected to grow at a compound annual growth rate of 49.8% [2]. Also, every smartphone provides an ASR interface, such as Apple Siri, to the user. Nevertheless, noisy environments are still challenging for ASR systems. For example, the keyword spotting (KSW) [3] accuracy of a commercial ASR system shows large degradation when noise increases, as shown in [4]. There are 1000 noisy samples represented as dots and are tested by a specific ASR system. It includes a linear curve fit with an R^2 value that shows word error rate is inversely proportional to SNR. The red line shows the word error rate without added noise.

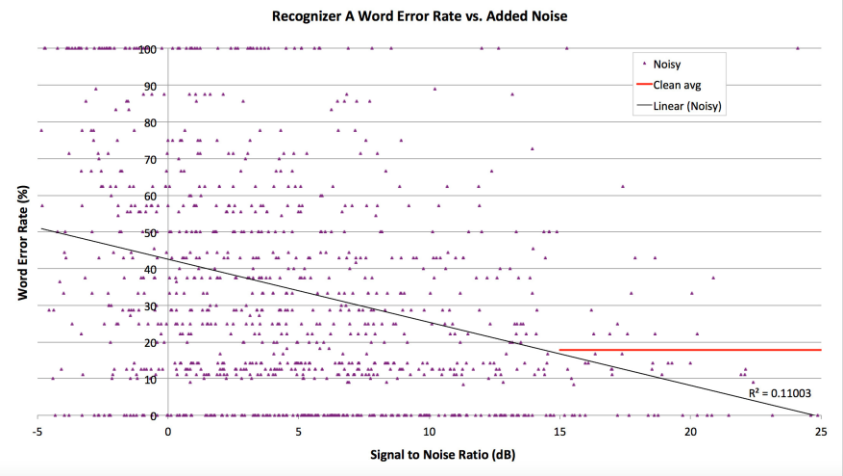


Figure 1-1. Word error rate versus SNR [4].

Beamforming is vital for the accuracy of ASR in noisy environments. Many commercial products, including Google Home, Amazon Alexa, and Apple Airpods, have multiple microphones and beamform. For example, a teardown of Amazon Echo (Figure 1-2) shows four microphones and two ADC chipsets (2 channels each). As a result of beamforming with a multi-channel input, the system can emphasize target speech coming from the desired direction while suppressing noise from other directions. On the other hand, the analog frontend linearity requirement in speech application is less stringent than that of RF frontend in wireless application because of the relatively high SINR of the target signal and the ease of high-linearity analog block design. As a result, the ASR system places beamformers after ADC in the digital domain

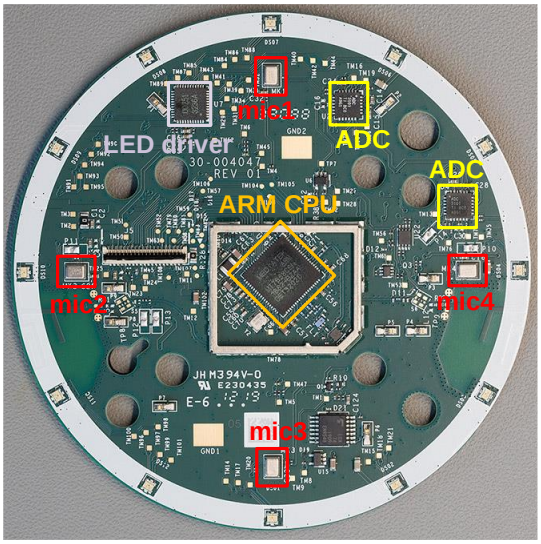


Figure 1-2. Teardown of Amazon Echo [5].

However, due to the unique characteristics of a speech signal system, the ASR frontend presents challenges to both ADCs and digital beamforming. First, the audio frontend ADC requires high SNR (>80dB) for a high-quality audio processing and speech recognition system. For example, Amazon Echo uses Texas Instruments TLV320ADC3101 dual sigma-delta ADCs to

digitize the analog inputs from four microphones [5]. Furthermore, battery-powered operation strictly limits power consumption. Also, as there is an increasing demand for highly integrated systems, the ADC circuit should adapt to more advanced technology – this is challenging for analog circuits due to low supply voltage, flicker noise, and device mismatch [6].

Second, the beamformer usually requires complex calculations since speech is a wideband signal ranging from 30 to 10kHz. For example, consonant (k, p, s, etc.) sounds reside primarily in the 2~4kHz frequency range, while the vowel sounds mainly lie below 1kHz [7]. Usually, the frequency range of 1~4kHz is critical for the intelligibility of the speech. On the other hand, most of the voice energy is in frequencies below 1kHz [7]. Hence, the signal processor becomes complicated to handle a multi-octave frequency range. Some beamformers separate the input signal into multiple bands using an FFT, then apply an independent narrowband beamforming algorithm to each band, and finally merge the outputs [8].

A noise-shaping ADC is a popular option for high SNR audio applications. A noise-shaping ADC pushes the quantization noise to out-of-band and filters out this quantization noise to achieve high SNR. A noise-shaping ADC can be implemented in two different ways: discrete-time and continuous-time. A discrete-time ADC, which is the dominant type in the market, operates with switched-capacitor circuit techniques. It is robust against PVT because the ratio of passive elements decides the filter coefficients. However, a discrete-time ADC needs an anti-aliasing filter before the discrete input sampling. Also, the required power consumption of the amplifier tends to be high due to capacitive sampling because the amplifier needs to charge a capacitor fast enough to achieve a certain level of accuracy.

On the other hand, a continuous-time ADC ([6], [9]) uses a continuous integrator at the input stage, resulting in an inherent anti-aliasing filter. Also, a continuous-time ADC tends to use

less power in the amplifiers than a discrete-time ADC since there is no capacitive settling. Therefore, recent research focuses on continuous-time ADCs due to the advantages over a discrete-time.

There are two categories of wideband beamformers: fixed beamformers and adaptive beamformers [9]-[11]. Examples of a fixed beamformer are the delay-and-sum [12][13] and the filter-and-sum beamformers [14][15], whose coefficients are fixed and independent of the input signal. However, when the noise conditions are changing, the performance of a fixed beamformer can degrade since the beamformer does not respond to the change.

On the other hand, an adaptive beamformer adjusts its beampattern depending on the target and noise situations. Well-known adaptive beamformers are the Minimum Variance Distortionless Response (MVDR) beamformer, Linear Constraint Minimum Variance (LCMV) [16] beamformer, and Generalized Sidelobe Canceller (GSC) beamformer [17]-[20]. These beamformers optimize their filter coefficients in real-time to minimize the beamformer output power under given constraints, such as unity gain for the signal coming from the desired angle. As a result, the beamformers can preserve the target signal while suppressing noise. Furthermore, some systems take advantage of DNNs combined with beamformers. [21][22] combines a conventional MVDR beamformer and DNN based direction of arrival (DOA) estimation, and [23]-[31] utilizes DNNs to learn filter coefficients from various input and noise situations.

1.2. Thesis Contributions

This thesis introduces three different fully-integrated ASR frontends. It investigates the following aspects: 1) effective and novel beamforming methods with the area and power-efficient implementations, 2) high-performance ADCs that are specialized for ASR frontends, and 3) synergy from co-design of the ADCs and the DSP.

In Chapter 2, we show: 1) the direct use of an SDM bitstream output can improve the steering accuracy, 2) and the combination of delay-and-sum (DAS) and constant-directivity beamforming (CDB) is effective for wideband speech beamforming. We describe our first fully integrated system ([12][13]), which harnesses the efficiency of DAS beamforming by combining it with CDB, and consumes 4mW. The design takes advantage of bitstream processing of the SDM outputs for beamforming with accurate steering. CDB facilitates DAS by restricting the bandwidth for different microphone configurations. Processing the Mel spectrum outputs with a DNN, the KWS accuracy in the presence of noise improves from 74% without beamforming to 93% with beamforming. However, this prototype is a fixed beamformer, so it cannot handle the varying noise conditions. Also, we found that operating frequency optimization can improve power efficiency.

Hence, Chapter 3 investigates: 1) design methods for combining bitstreams output of SDM ADCs with adaptive beamforming, 2) improving noise-suppression performance compared to the system in Chapter 2 through adaptive beamforming, 3) techniques for area/power-efficient hardware implementation of the beamformer. The prototype ([18][19]) introduces a fully integrated system with a 4-channel input adaptive beamformer with 0.65mW power consumption. It combines fast switching bitstream output of continuous-time SDMs (CT SDMs) and DAS to achieve a high accuracy steering angle. A time-domain RGSC adaptive beamformer can effectively suppress the varying noise input. Also, it shares hardware and optimizes a DSP clock speed for efficient implementation. As a result, the beamformer improves speech recognition

accuracy in noisy conditions from 64% to 90% using DNN trained with noisy speech. On the other hand, this prototype still consumes significant power due to the complex blocking matrix (BM) calculation and full-performance mode (both in ADC and beamformer) regardless of the input SNR condition. Also, it likely shows signal distortion because of the sensitivity in coefficient adaptation control.

Finally, in Chapter 4, our main contributions are: 1) a novel adaptive beamforming algorithm with DOA tracking by utilizing the direct output of ADCs, 2) a multi-mode beamformer with novel multi-mode ADCs for a low-power operation that takes advantage of the input signal situation, and 3) multi-mode control schemes. We demonstrate a fully integrated system ([32]) with a 4-channel multi-mode ADCs and a greedy blocking matrix beamformer consuming 157 μW in total. First, we replace the BM from Chapter 3 with a newly proposed greedy BM (GBM) to reduce signal distortion and automatically adjust the input steering error. Also, the beamformer has two modes to deal with varying noise conditions: fully adaptive beamforming (10μW) and DAS only beamforming (49μW). Second, we design continuous-time noise-shaping SAR (CT NSSAR) that operates in three modes: CT NSSAR (80dBA/12μW), NSSAR (65dBA/5.8μW), and SAR (40dBA/1.5μW) modes. In this way, our proposed beamformer can optimize its power consumption depending on the varying signal and noise conditions. It improves KWS accuracy from 54% to 83% under word interference using DNN trained with noisy speech.

Chapter 2. Frequency-Selective Bitstream DAS Beamformer and Feature Extractor

2.1. Motivation

Automatic speech recognition (ASR) has become practical thanks to the progress in deep neural networks. However, acoustic beamforming with multiple microphones is essential to suppress environmental noise and interference in realistic application scenarios. Further challenges are the required very high dynamic range (>80dB) and the multi-octave frequency range of speech. The very-wide frequency range necessitates extensive DSP for frequency-dependent beamforming and feature extraction.

This work [12][13] focuses on three issues of conventional beamformers: (i) the power-hungry multiple FFT/IFFT operations needed for wideband beamforming, (ii) the necessity of fine time resolution of the delay line for delay-and-sum (DAS) operation for accurate steering, and (iii) the limitations of the narrowband characteristic of DAS.

A conventional wideband beamformer (Figure 2-1 left [31]) requires (i) an array of high-resolution ADCs along with decimation filters, (ii) multiple FFTs, weighting, and IFFT for wideband beamforming (iii) windowing, FFT, filtering, and energy calculation for feature extraction [33][34]. [34] describes an efficient hardware approach for feature extraction, but its ADC performance is limited to ~48dB SNR, and it cannot be combined with beamforming due to its data processing scheme.

---

*This work was done in collaboration with Seungjong Lee and John Bell. The author's main contribution is digital synthesis, measurement, and speech recognition MATLAB simulation.
The text and figures are based on [12][13].

In the proposed system (Figure 2-1 right), the single-bit quantizer outputs of the eight continuous-time SDMs directly feed to the beamformer, removing the standard requirement for decimation filtering. Also, the time-domain feature extractor removes multiple FFT/IFFT blocks.

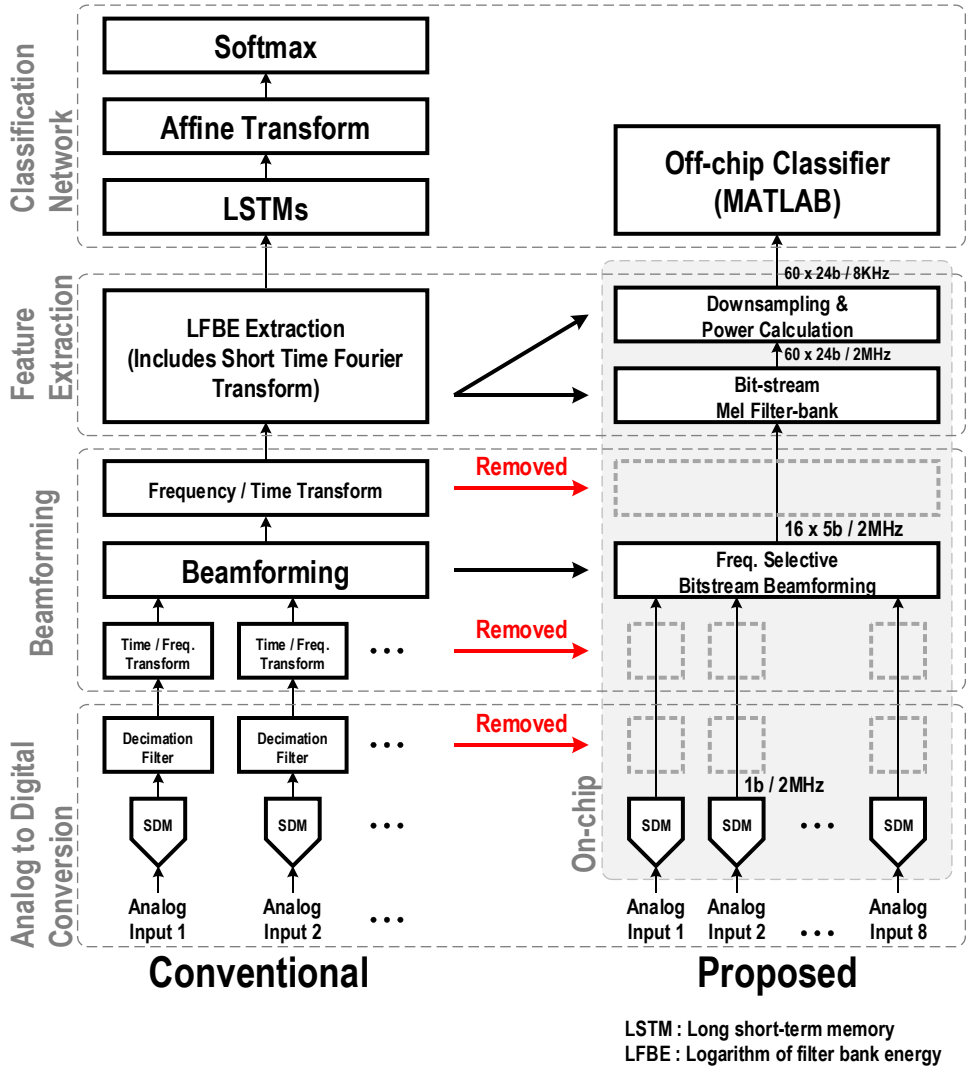


Figure 2-1. Overall ASR diagram of a conventional system, based on [31] (left) and proposed system (right)

DAS beamforming is simple and effective for narrowband acoustic beamforming in sonar and ultrasound imaging. As shown in Figure 2-2, DAS beamforming sums delayed signals from multiple microphones to reinforce the desired signal and disperse or attenuate interferers constructively. DAS beamforming improves the in-band SNR due to noise in the ADC and

frontend because the desired signal is correlated between channels, while this noise is uncorrelated. However, the delay resolution is a challenge with DAS, especially for digital DAS beamforming, because the ADC sampling rate determines the time resolution. For example, the Nyquist sampling rate is 16kHz for a speech bandwidth of 8kHz, leading to a time-resolution of 62.5μs, which is too coarse for small-aperture microphone systems. With a 1-inch spaced linear microphone array and a time resolution of 62.5μs, DAS beamforming is limited to a steering resolution of about 60 degrees – this is too coarse to track a speaker in a room. [35] uses a fractional delay cell to overcome this time resolution problem, but it takes complex calculations.

To solve the time resolution problem, we utilize the SDM's direct output by taking advantage of oversampling. This approach performs DAS on the modulator bitstream (without decimation), thereby exploiting the time-resolution of the over-sampling clock. As a result, the proposed system uses a 3.4μs delay step instead of 62.5μs, providing enough time resolution for accurate steering.

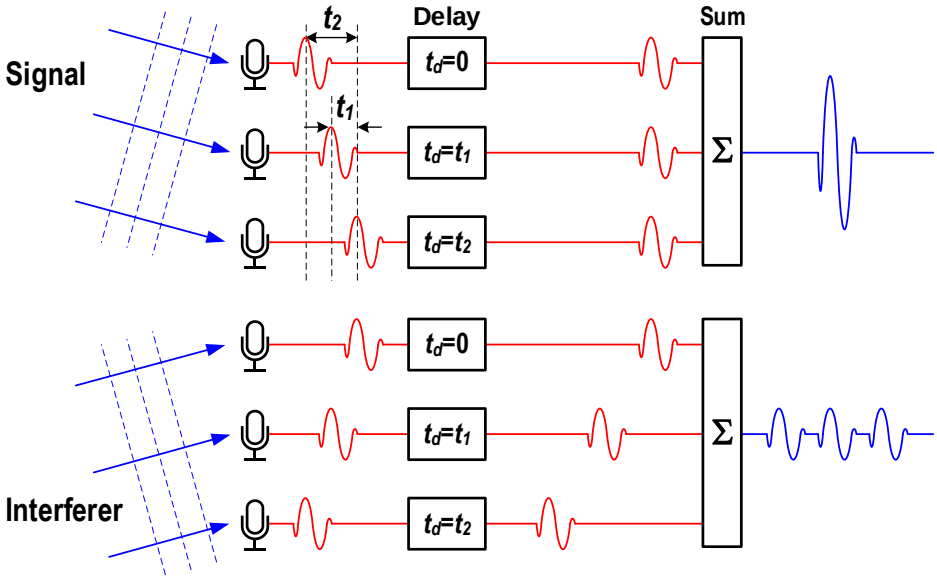


Figure 2-2. Delay-and-sum beamforming enhances the desired signal (top) and attenuates an interferer (bottom).

Next, despite the advantages of oversampling, DAS is unsuited to speech because it only works for narrowband beamforming. For example, DAS in ultrasound imaging [36]-[38] operates with a single frequency band covering a ±20% frequency range. However, speech recognition systems typically consider a frequency range spanning 7 octaves and are interested in tens of frequency bands. If the microphone distance is large compared to the signal wavelength, aliasing occurs, and it causes some interferers to leak into the output. On the other hand, if the microphones are too close together, then the phase difference between each element is too small to generate constructive and destructive interference. As the microphone array configuration and wavelength determine the beamforming characteristics and the beampattern, DAS can only deal with narrowband signals.

Hence, the proposed system realizes constant-directivity beamforming (CDB) characteristics by adopting frequency-selective beamforming and corresponding multi-configurations of microphones to solve the narrowband limitation of DAS.

This work includes the following features: (i) frequency-selective bitstream beamforming, (ii) bitstream Mel frequency-band feature extraction, and (iii) an array of efficient continuous-time SDMs without area/power-intensive decimation and FFT/IFFT.

2.2. System Implementation

2.2.1. Overview

This work takes advantage of the simplicity of DAS in frequency-selective beamforming. By combining CDB and DAS, we restrict DAS operation to relatively narrow frequency ranges while optimizing the microphone placement in those frequency ranges. Furthermore, we argue that merging spectrogram generation with beamforming is essential for the efficient combination of CDB and DAS. Typically, in ASR systems, a Short-Time-Fourier-Transform (STFT) generates the spectrogram from the beamformed signal. Instead, our approach combines a bandpass filter bank with energy-detectors to replace the SFTF. In addition to saving energy over the STFT approach, we will see that a crucial advantage of the filter-bank is that it facilitates the appropriate combination of CDB and DAS.

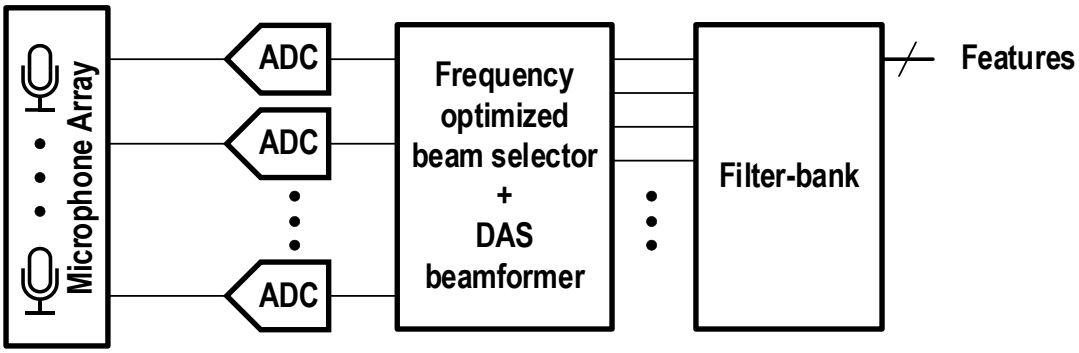


Figure 2-3. Simplified system block diagram.

Figure 2-3 is a simplified depiction of how we combine CDB, DAS, and spectrogram generation. A dedicated ADC digitizes each microphone signal. The single-bit output of each ADC output feeds a delay line, and in turn, the delay lines feed a bank of bandpass filters. The delay lines ensure a fine delay resolution. The input to each bandpass filter in the filter-bank is the weighted sum of selected taps from selected delay lines. In this way, each frequency band has its

own DAS beamforming configuration. Furthermore, setting a weighting to zero excludes a particular microphone from the beamforming for that frequency band. Therefore, the weightings allow each frequency band to have a unique microphone configuration, thereby facilitating CDB. The filter-bank is an array of bandpass filters with an approximate Mel-frequency (i.e., log) spacing. Sliding window energy detectors form the spectrogram at the filter outputs.

2.2.2. System Architecture

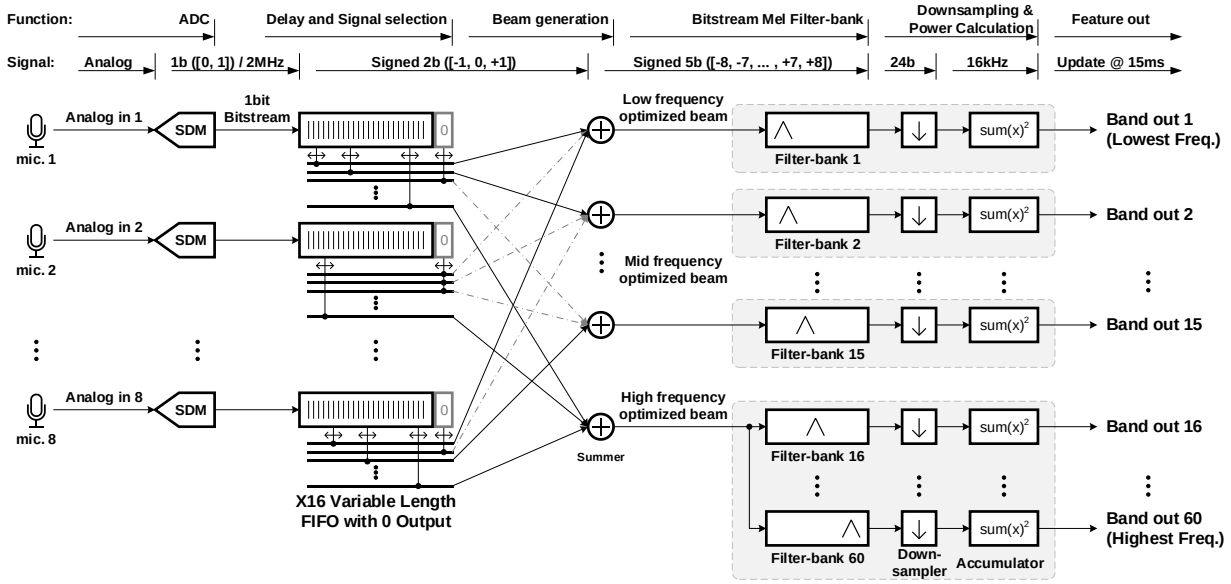


Figure 2-4. Detailed system diagram showing frequency-dependent DAS, the Mel filter bank, and the band energy calculation.

Figure 2-4 shows a system-level diagram of the proposed system. Flexible frequency-selective beamforming is essential for sophisticated beamforming techniques, including CDB and 2D-array beamforming. The eight bitstream outputs of the SDMs feed eight bitstream delay lines. Sixteen summers tap and sum programmable positions of the delay lines to form sixteen different beams. The 15 lower-frequency feature filters (i.e., 1-15) are each fed independent beams. The higher band filters (i.e., 16-60) share a single beam. An advantage of the proposed structure is that

the incremental cost of additional microphones is low because there are no individual decimators or FFT units.

2.2.3. CDB Array configurations

The proposed system uses two different microphone configurations for constant-directivity beamforming of high and low frequencies. Although our prototype supports 16 simultaneous frequency-dependent configurations, we demonstrate that two configurations provide good performance.

When designing the microphone array, it is best to make the sensor spacing as large as possible to better beamform low-frequency signals. However, the large wavelengths at low frequency suggest impractically large spacings; for example, the half-wavelength for a 600Hz tone is 11 inches. Therefore, to demonstrate a practical solution, we limit the array area size to fit within a 5-inch diameter, similar to commercial speech-beamforming products such as Amazon Echo and Google Dot.

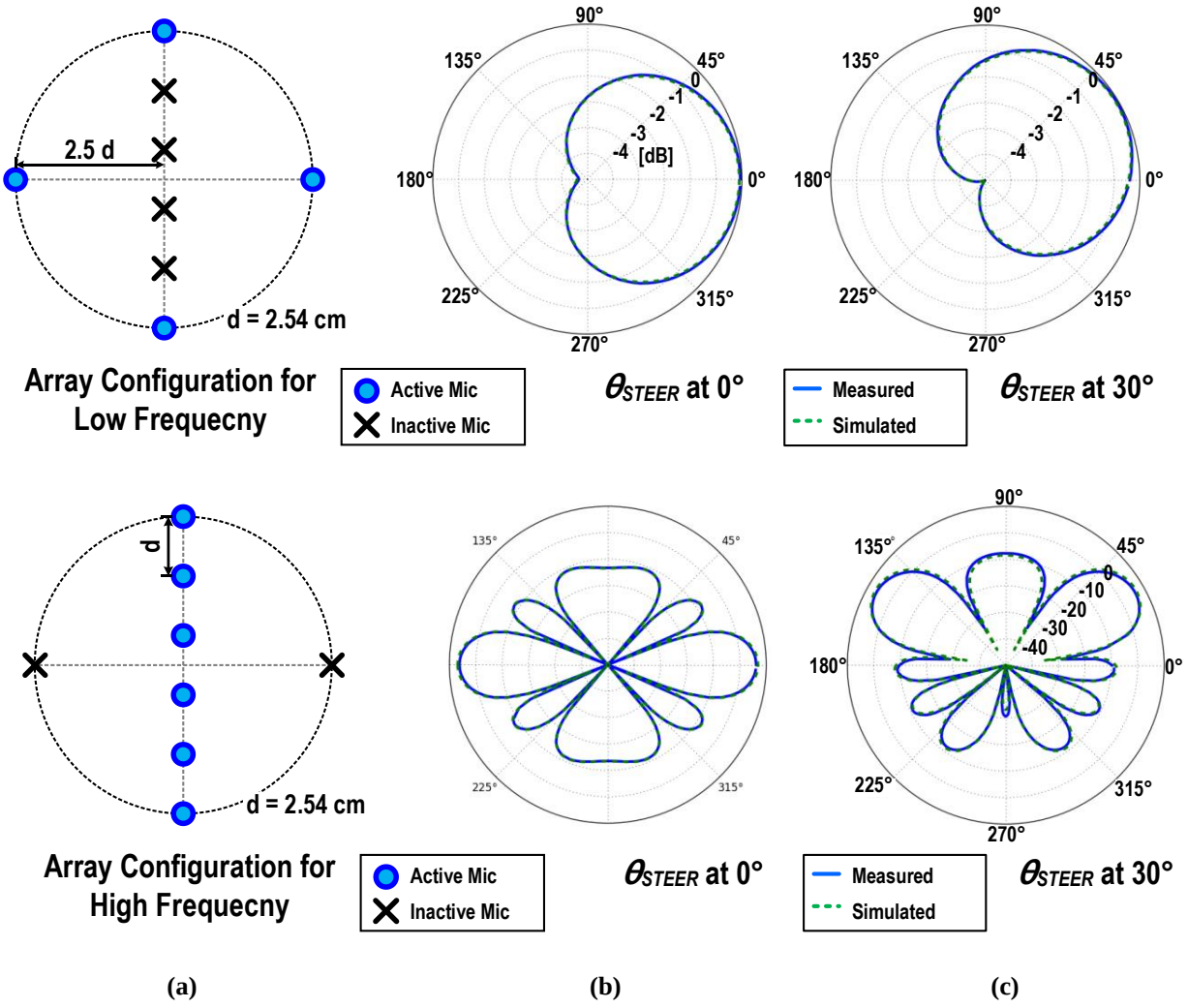


Figure 2-5. Fig. 6. (a) Microphone array configurations, (b) simulated beampatterns at 597.7Hz, and (c) simulated beampatterns at 6kHz.

Figure 2-5 (a) shows the microphone configurations for the two frequency ranges. We use a Uniform Linear Array (ULA) for high-frequency beamforming since it has high directivity. We use a Uniform Circular Array (UCA) for low frequencies, which offers the largest distance between sensors and enough microphones for a good SNR. The two configurations share the microphones located at the top and bottom. Figure 2-5 (b) and (c) show the simulated beampatterns for a beam steered to 0 degrees.

Beginning with high-frequency inputs, we see that at 6kHz, ULA provides strong directivity in the desired direction while attenuating the most prominent side lobes by 12.5dB. In contrast, UCA only attenuates the side lobes around ±90 degrees by less than 1dB, which means there is little directivity. The situation is the opposite for lower frequencies. At 597.7Hz, ULA provides a near-uniform response with only 1dB of rejection at ±90 degrees. The four-element UCA provides the classic cardioid response at low frequencies. Unlike ULA, UCA attenuates signals from the rear providing a 4.5dB better noise rejection compared to ULA. Although the distance between sensors is only 16% of the wavelength, directivity and SNR are better than a two-microphone linear configuration. A critical advantage of the cardioid response is the strong rear rejection – this is vital for suppressing echoes and reverberation in practical scenarios. In conclusion, using the different microphone configurations at different frequency ranges improves directivity and signal quality.

2.3. Circuit Implementation

2.3.1. Digital Signal Processor

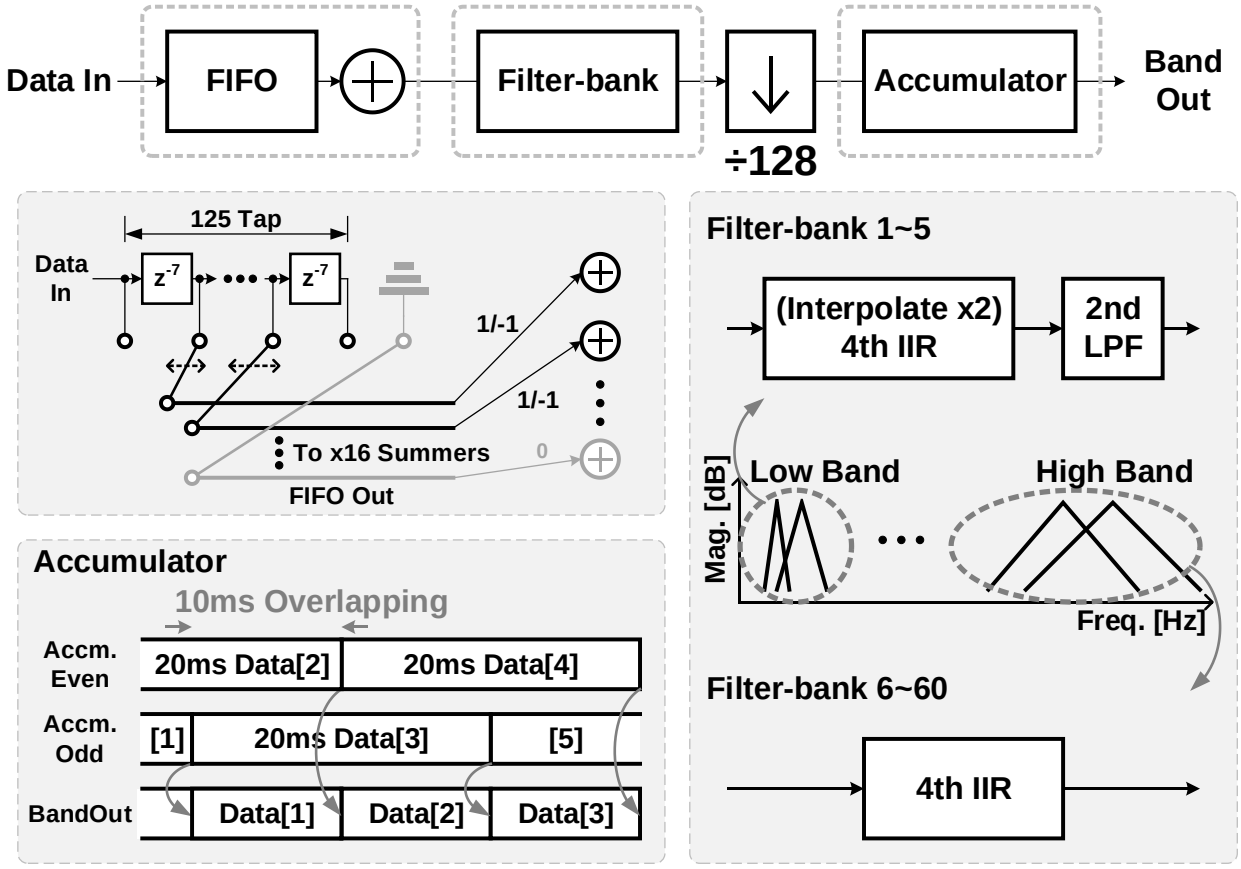


Figure 2-6. Details of a bitstream delay line, 60 channel filter-bank, and energy accumulator.

Figure 2-6 shows the digital processor in detail. A beamforming slice for each SDM passes the 1-bit SDM output through a 125-tap delay line. Each tap provides 3.4μs of delay, which corresponds to a beamforming resolution of 2.6 degrees for a 1-inch linear microphone array. The nominal delay line output values are -1 and 1; however, an output can be fixed to 0 so that a microphone does not contribute to a formed beam. The 16 full-rate 5-bit beamformed signals feed the filter-bank. The high-band filters are simple 4th-order IIR filters. Due to the narrow transition band and the limited coefficient precision, the high oversampling rate makes the low-frequency filters more challenging to implement. We solve this challenge with a 4th-order interpolation IIR

structure and remove the high-frequency images created by the interpolation with a low-order low-pass filter.

The filter-bank outputs are down-sampled to 8kHz and sent to a windowing energy calculator (Accumulator in Figure 2-6). The energy calculator determines the sum of the signal squared within two parallel windows. The windows overlap in time, similar to the way the windows of a Short-Time Fourier Transform overlap. The energy in each window is then passed to the ASR algorithm.

2.3.2. Continuous-time Delta-sigma Modulator

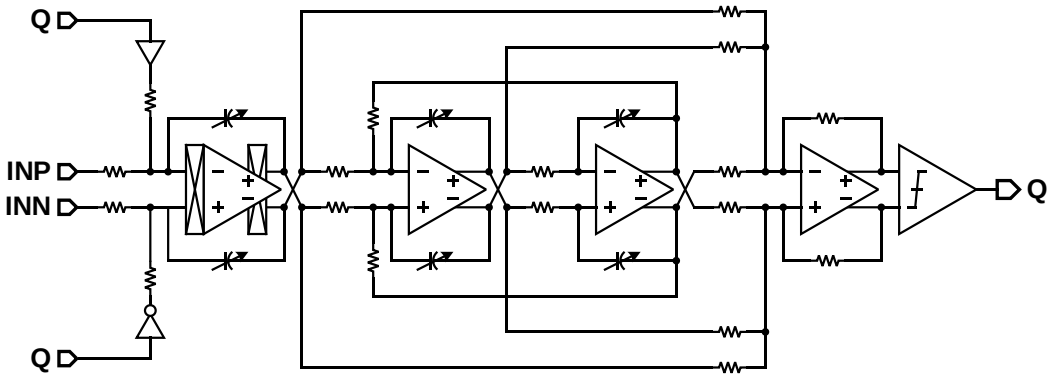


Figure 2-7. Third-order continuous-time SDM with chopping and 85dBA SNDR.

Figure 2-7 shows a schematic of the continuous-time SDM. The measured SNDR of a single SDM is 85dBA. The overall system benefits from the array gain so that the entire array with eight parallel SDMs has an SNR that is 9dB higher. The modulator is a 1-bit 3rd-order feed-forward architecture. The sampling frequency of the modulator is 2.048MHz for an 8kHz bandwidth, corresponding to an oversampling ratio of 128. In the first integrator, chopping suppresses flicker noise. A 3-stage class AB amplifier improves current drive, efficiency, and linearity. The SDM occupies 0.054mm^2 and dissipates 91µW.

2.4. Measurement Results

The prototype is fabricated in a 40-nm general-purpose CMOS process. The total active area is 1.1mm^2. Figure 2-8 shows the layout and identifies the main blocks. Since the digital circuitry operates at low speed, the digital supply voltage is 0.55V. The analog circuitry operates under a 1.0V supply for better noise performance.

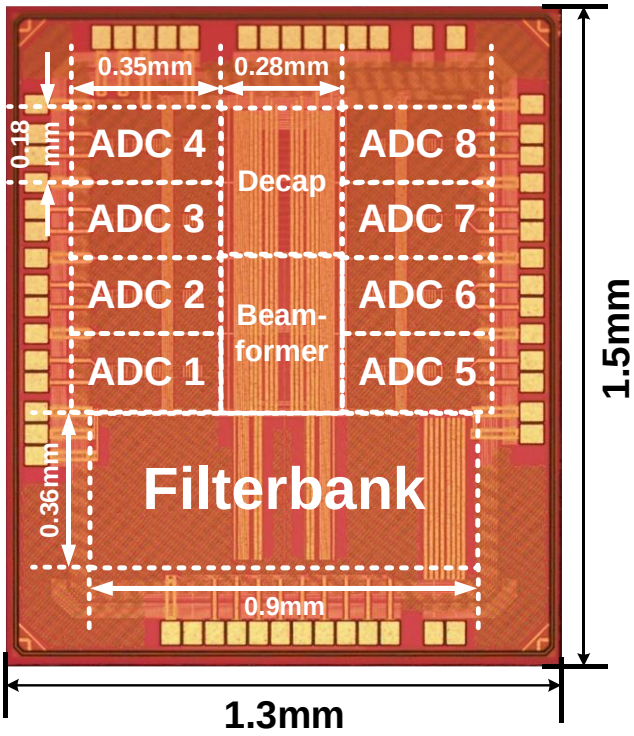


Figure 2-8. Die photo.

2.4.1. Test Setup

The test setup facilitates high-accuracy measurement of beampatterns as well as characterization of keyword spotting accuracy with and without background noise (Figure 2-9). An 8-channel 24-bit audio DAC (Cirrus Logic CDB3485) provides eight audio inputs to the prototype IC, emulating a microphone array. The DAC channels can carry independent signals; however, an essential advantage of the 8-channel DAC is that the phase relationship between

channels is well controlled. An Opal-Kelly XEM 7001 FPGA board controls the eight-channel DAC. Eight low-noise single-to-differential amplifiers (Analog Devices ADA4940) convert the single-ended DAC outputs to differential signals. The FPGA also reads the prototype IC's spectrogram outputs and sends this information to a PC that performs the final stages of keyword spotting in MATLAB.

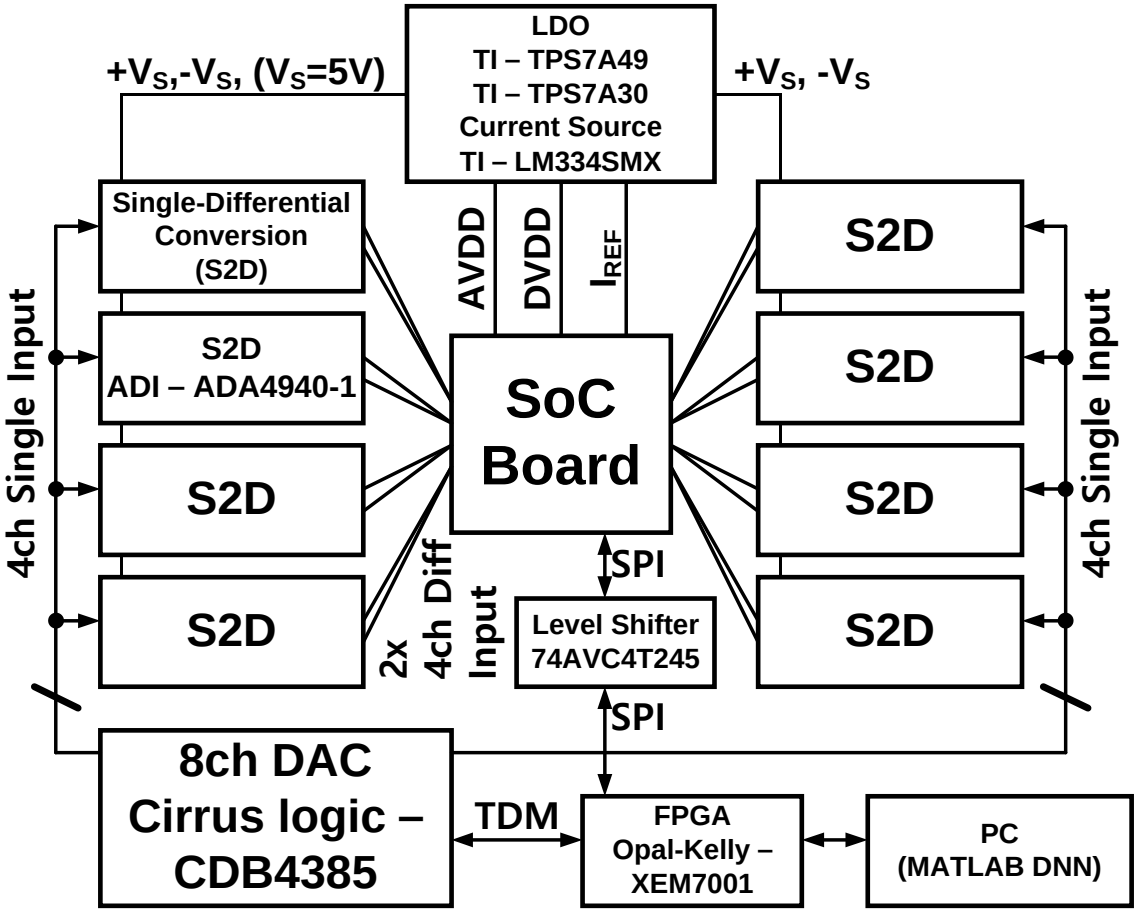


Figure 2-9. Block diagram of the test setup.

2.4.2. Measured Beampatterns

Our microphone array combines cardioid and linear configurations (Figure 2-10) to optimize high-frequency and low-frequency performance. For high frequencies, the center six microphones form a linear beamforming array. For low frequencies, the outer four microphones operate in a cardioid configuration. Cardioid beamforming also provides backside rejection. Figure 2-10 plots the measured beam patterns for two different frequencies and two different steering angles.

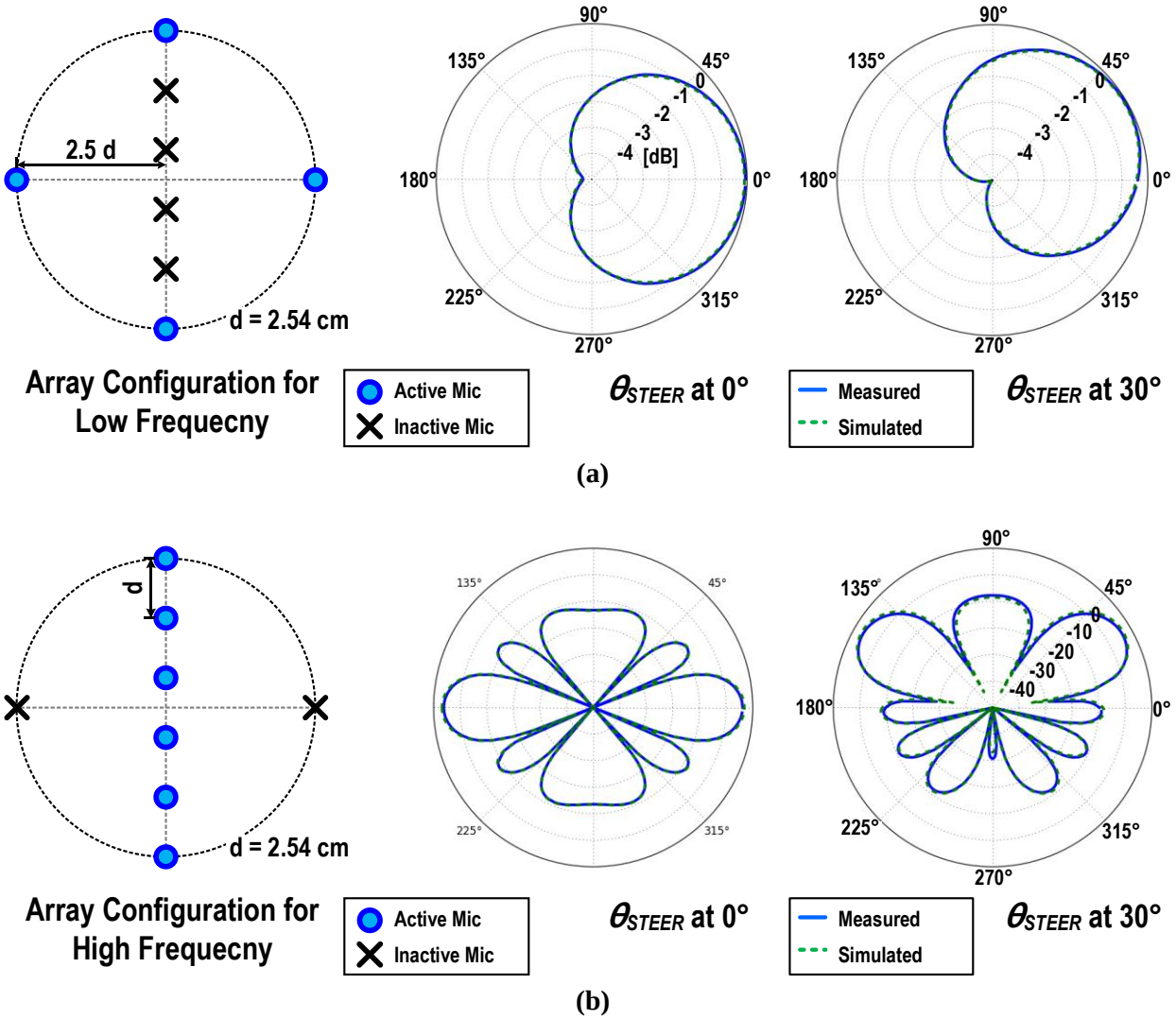


Figure 2-10. Array configuration and measured beam patterns for steering angles of 0 and 30 degrees for (a) high frequency (6kHz) and (b) low frequency (597.7Hz). The measured beam patterns are near-identical to the simulated ones.

2.4.3. Speech Recognition Test

2.4.3.1 Training DNN without Noisy Data (Measurement)

We verify the speech recognition performance of our prototype with the Tensorflow speech dataset. In testing, an 8-channel 24-bit audio DAC emulates an 8-element microphone array. We apply the MATLAB deep learning toolbox to implement an off-chip DNN. The dataset [39] consists of 720 utterances of 8 words, 1440 unknown words, and 1600 samples of background noise. We divide the dataset into training and validation samples with an 8:1 ratio. The measured recognition accuracy of our prototype without noise is 95%.

To demonstrate the advantages of beamforming, we also measure recognition accuracy in the presence of a noisy interferer, both with and without beamforming. With the 8-microphone configuration in Figure 2-11, we place the speaker at 0° and a random noisy interferer at 130°. Spectrograms in Figure 2-12 plot measured features for 1sec of speech with interference without and with beamforming. Beamforming improves the recognition accuracy from 74% to 93%. Figure 2-13 shows its confusion matrix.

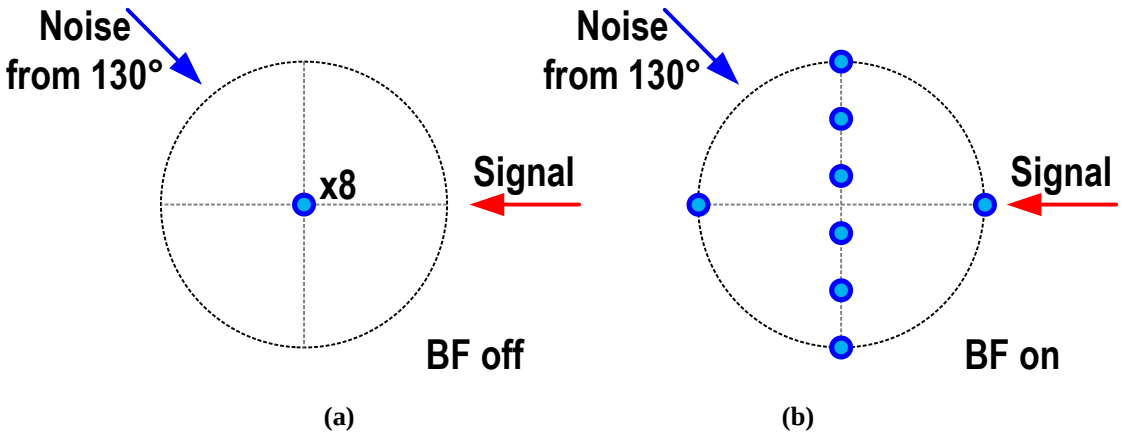


Figure 2-11. Microphone configurations, (a) without, and (b) with beamforming.

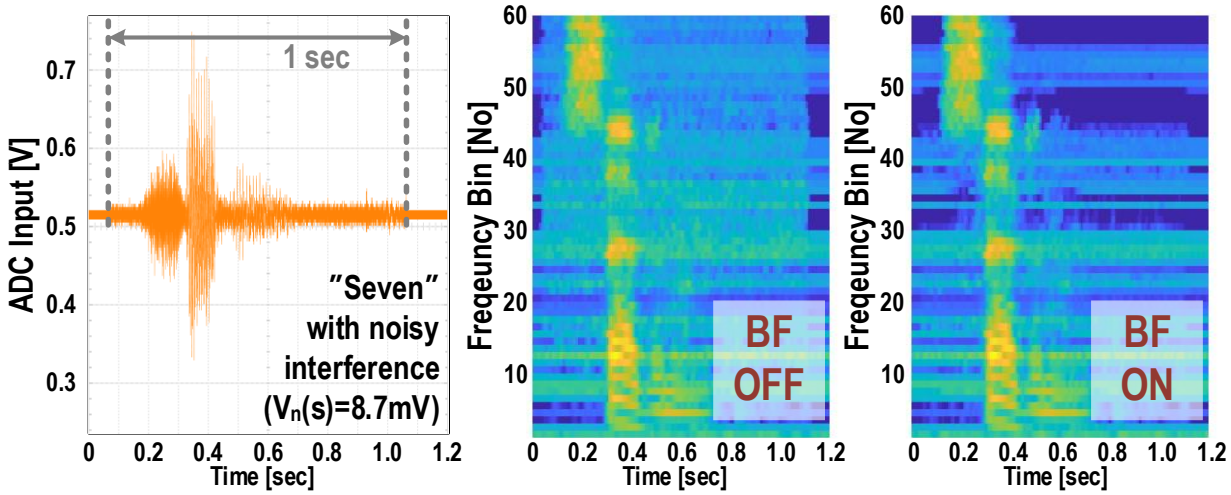
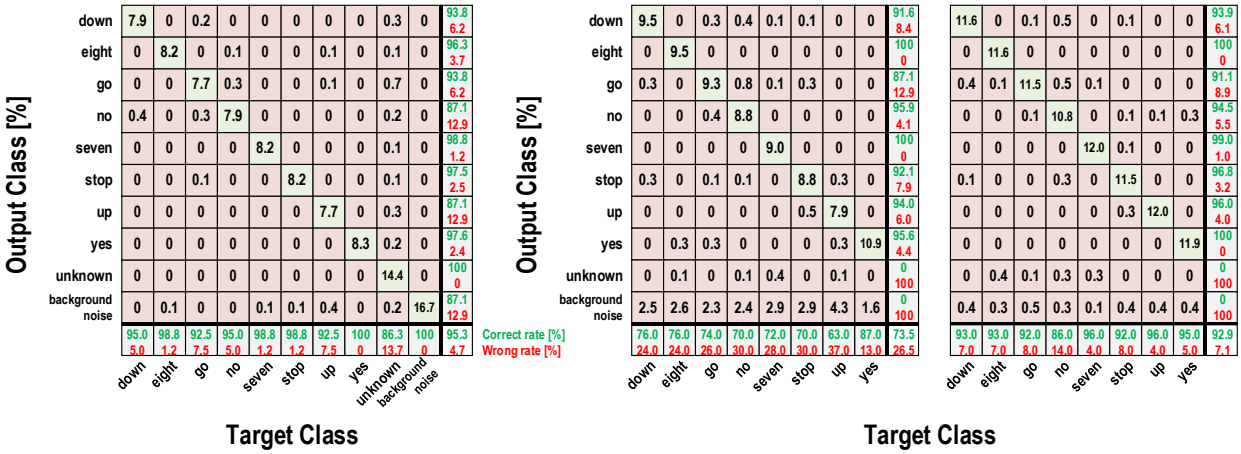


Figure 2-12. (left) Measured noisy speech waveform, (right) Beamforming improves the measured spectrogram (i.e., BF off vs. BF on).



(a) **(b)** **(c)**

Figure 2-13. Confusion matrix (a) for noiseless speech, (b) with noise and without beamforming, and (c) with noise and with beamforming enabled.

2.4.3.2 Training DNN with Noisy Data (MATLAB Simulation)

Although training a DNN with a noisy input improves classification accuracy, our MATLAB simulations highlight the benefits of beamforming. We first consider a white noise as an interferer and later consider random speech as the interferer. We set the training condition to be the same as the test condition, and the DNN is trained with added white noise (15.5dB SNR). If the input SNR is 15.5dB for a white noise interferer, the simulations show no performance degradation even without beamforming (92.8%). However, the classification accuracy significantly decreases to 75% without beamforming in the presence of a stronger interferer (i.e., 6dB SNR). In this case, beamforming improves the accuracy to 81.3%, suggesting the benefit of beamforming in unpredicted noise conditions. The advantage of beamforming is even more significant when the DNN is trained without noise. In this case, beamforming improves accuracy from 59.2% to 79.9% for an input SNR of 6.0dB.

Beamforming is even more effective if the interference is random speech. Training the DNN with white noise does not improve the classification accuracy if the interference is random speech, but beamforming significantly increases accuracy. If the desired signal and the random speech interferer have a power ratio of 15dB, then regardless of the DNN training condition, beamforming improves the classification accuracy and enables 90% accuracy. With more substantial interference (5dB ratio), beamforming improves accuracy by 13% to 77% when the DNN is trained both with or without noise. These simulations show that beamforming is beneficial in practical speech recognition scenarios.

We also use MATLAB simulations to investigate the effect of the signal's direction of arrival (DOA) and the interference on the recognition accuracy. We use the same DNN training with a noisy signal (15.5dB SNR with white noise). We first sweep the input signal's arrival direction from 0 to 360 degrees with no interferer (Figure 2-14 (a)). As expected, the accuracy

does not degrade thanks to the fine resolution of the DAS beamformer. Figure 2-14 (b) shows the same signal DOA sweep but with a large white noise interferer at 130 degrees (6dB SNR). The accuracy degrades when the noise and input signal come from the same direction since the beamformer cannot separate the noise and the signal.

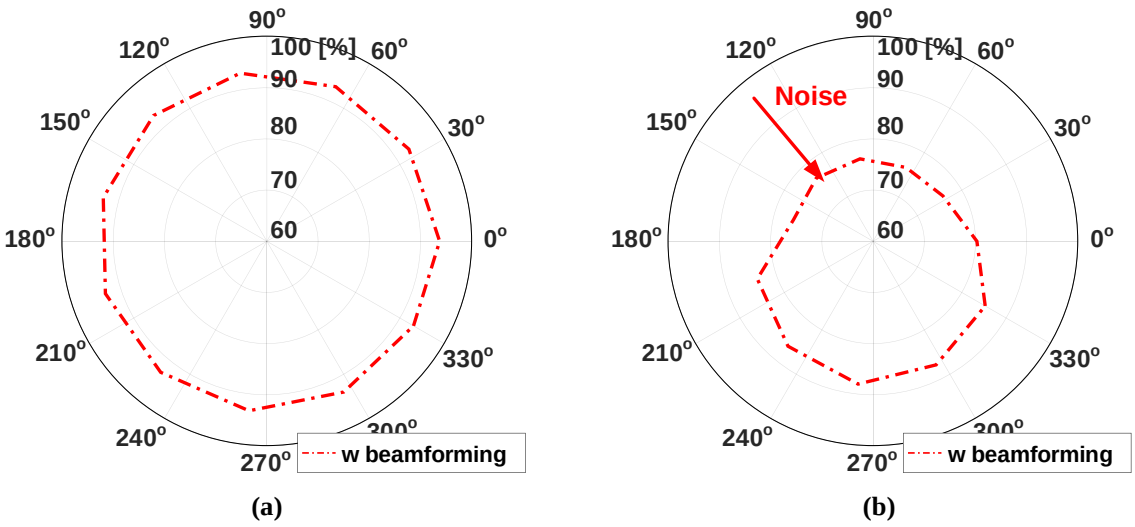


Figure 2-14. Polar plot showing classification accuracy versus DOA of the input signal: (a) without noise, and (b) with noise (6dB SNR) from 130 degrees.

Next, we consider the effect of the DOA of the interference, again using a DNN trained with a 15.5dB SNR white noise dataset to classify the simulated spectrogram. Figure 2-15 shows a polar plot of the classification accuracy versus the DOA of the interference: (a) with white noise interference and (b) with random speech interference. We fix the DOA of the desired signal and sweep the DOA of the interference. Figure 2-15 (b) highlights the advantages of beamforming in the presence of interfering speech.

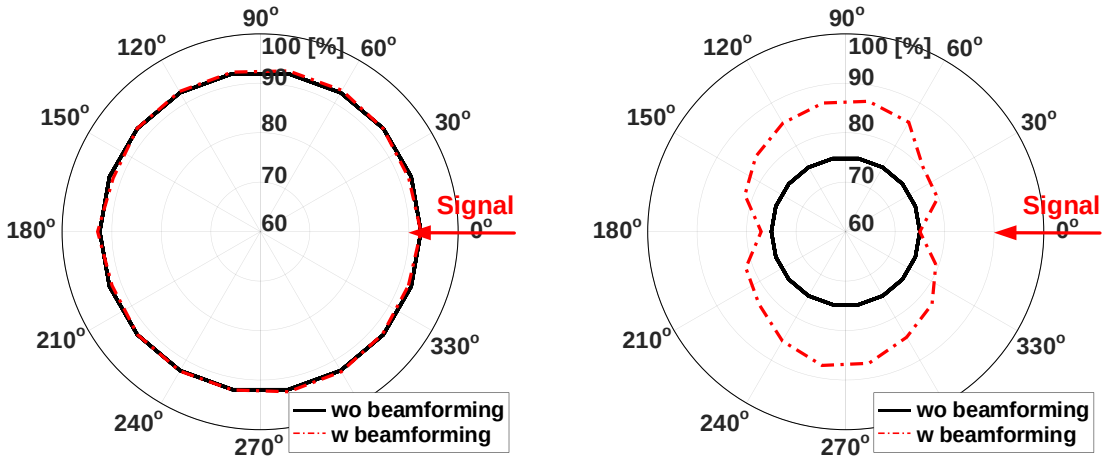


Figure 2-15. Polar plot showing classification accuracy versus DOA of the interference: (a) with white noise (15.5dB SNR), (b) with random speech interference (15dB signal power ratio). The DOA of the input signal is fixed at 0 degrees.

2.4.4. Power Breakdown

Figure 2-16 shows the power distribution. The total power consumption is 3.95mW. 74% of this power is from drain-source leakage. Leakage is high even with a low supply due to the use of the general (low Vth) process. Simulations show that a low leakage process would greatly reduce leakage and power consumption. The simulations show that the leakage power decreases to ~5µW; however, a low leakage process requires an increased digital supply (0.7V) [19] and a simulated 80µW increase in dynamic digital power consumption.

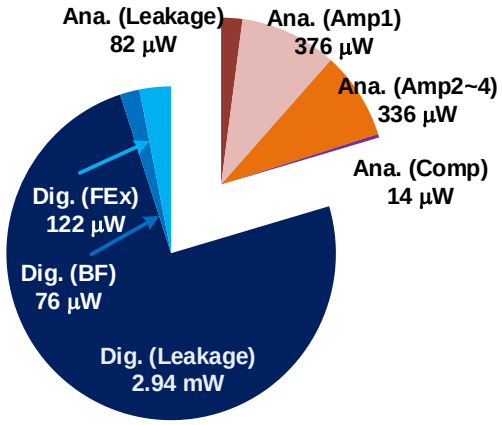


Figure 2-16. Power breakdown. The BF (i.e., beamformer) power includes the FIFOs and summers. The FEx (i.e., feature extractor) includes the filter-banks and energy calculators.

Chapter 3. RGSC Beamformer with Feature Extractor

3.1. Motivation

The prototype in Chapter 2 takes advantage of the simplicity of delay-and-sum beamforming (DASBF) for low-power applications. However, a fixed beamformer does not effectively suppress varying noise, limiting practical applications in the real world, where noise is usually not stationary. In this chapter, we present an adaptive beamformer that suppresses varying noise sources.

To begin with, we review DASBF. DASBF uses time-alignment and summation (e.g. [12]). Figure 3-1 describes the principle of DASBF. When the signal comes from a certain direction, each microphone will receive the same signal with different delays. As shown in Figure 3-1 top, the 1st ADC receives the signal first, and Mth ADC receives the last. If we set $t_{d,1\text{-}M}$ properly, we can align the phases of each channel's signal. Summing the aligned signal will result in an enhanced signal. Next, we assume $t_{d,1\text{-}M}$ are fixed to these values and consider noise coming from another direction. In this case, the given $t_{d,1\text{-}M}$ do not align the noise since the relative phases of received signals are different from the previous ones. Hence, signals are not aligned after the delay, and summing the un-aligned signal will attenuate its magnitude. As a result, the DASBF can enhance or suppress the signals depending on its direction of arrival (DOA) by setting $t_{d,1\text{-}M}$. However, as mentioned earlier, a critical drawback is that DASBF can only suppress noise from a fixed direction, making it ineffective for practical scenarios with multiple constantly changing noise sources.

*The text and figures are from [18][19].

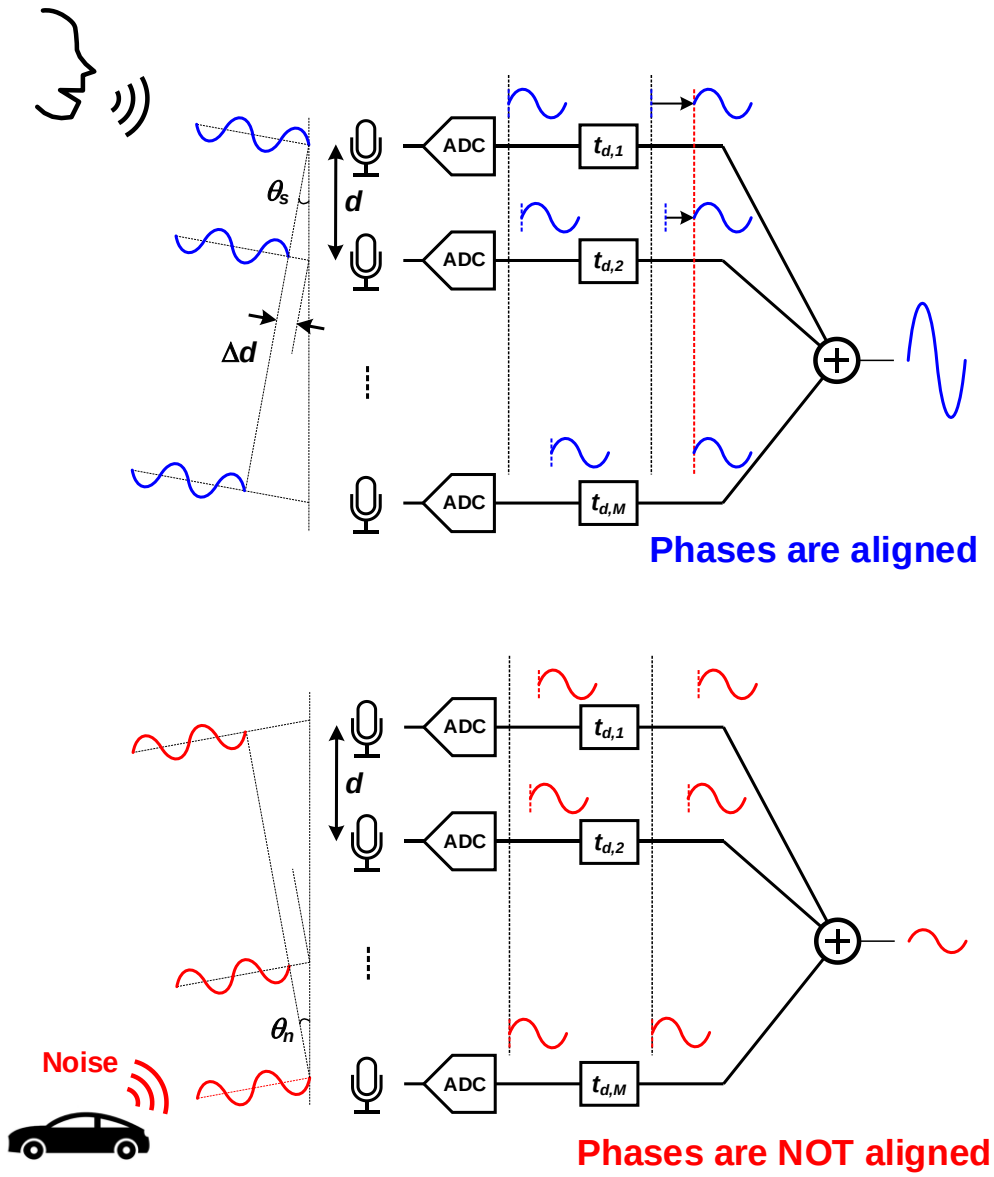


Figure 3-1. The principal of delay-and-sum beamformer.

On the other hand, adaptive beamforming (ABF) solves the limitation of DAS by automatically and adaptively suppressing noise from multiple, varying sources (Figure 3-2). However, high power and large area impede the implementation of conventional ABF. Another challenge is that high angle accuracy is crucial for ABF to avoid severe distortion of the desired signal [17]. We address these challenges by combining the robust generalized sidelobe canceller

(RGSC) algorithm [17] with bitstream processing for accurate steering and low-power ABF [18]. Hardware sharing and an optimized DSP clock rate further reduce power and area. The prototype system [18][19] includes multi-channel digitization, beamforming, automatic noise suppression, and feature extraction for a robust sub-mW single-chip speech-processing frontend.

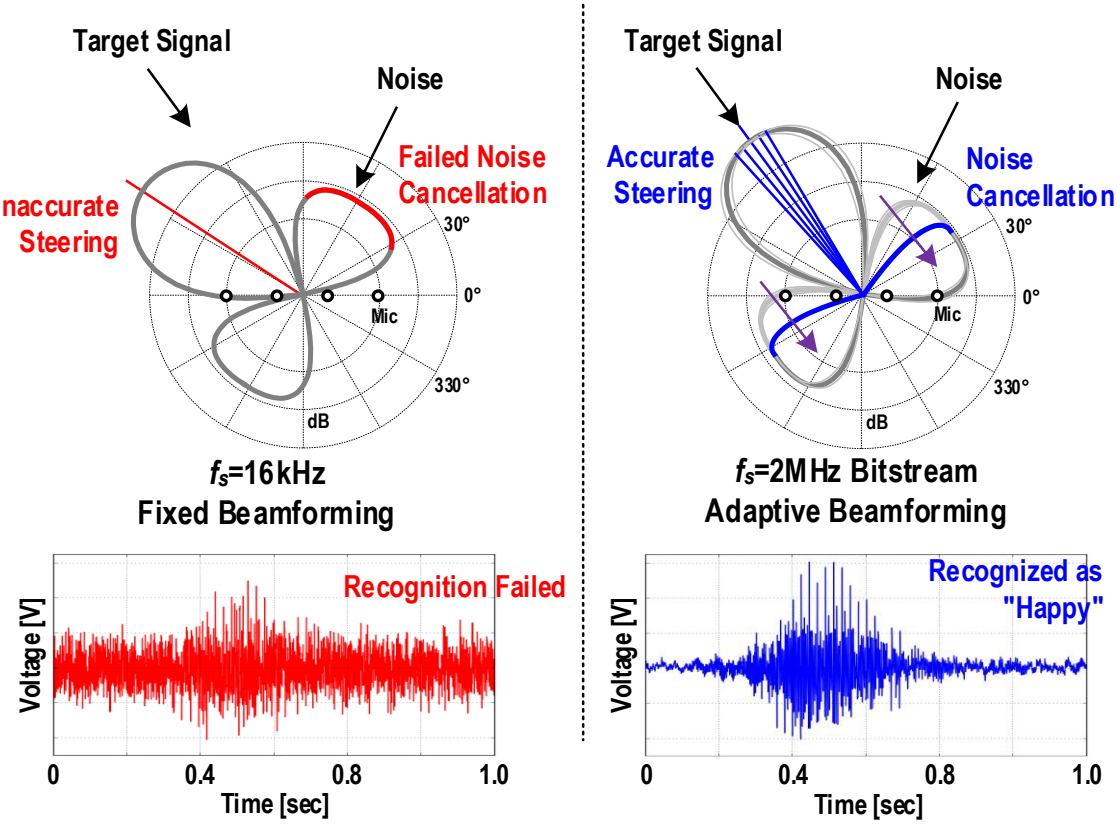


Figure 3-2. (left) DASBF cannot adapt to changing noise direction and has a limited angular resolution, and (right) bitstream ABF automatically places nulls in the noise directions and has high angular accuracy.

Figure 3-2 compares ABF with conventional DASBF, using polar plots to indicate directional gain. A serious deficiency of DASBF is that its beamforming nulls are fixed and therefore do not suppress interfering noise sources. In comparison, ABF automatically adapts null locations to suppress noise sources optimally. A further challenge is that DASBF with a conventional 16kS/s ADCs (i.e., 2x audio bandwidth) cannot accurately select the target signal. Our approach combines ABF with bitstream processing for optimal suppression of varying noise

sources and highly accurate target selection. The prototype includes four ADCs, ABF processor, and a frequency-domain speech feature extractor in a 40nm LP CMOS die. The entire noise-canceling and feature-extraction system consumes 0.65mW and improves speech recognition accuracy in the presence of two noise sources from 64% to 90%.

3.2. System Implementation

3.2.1. System Overview

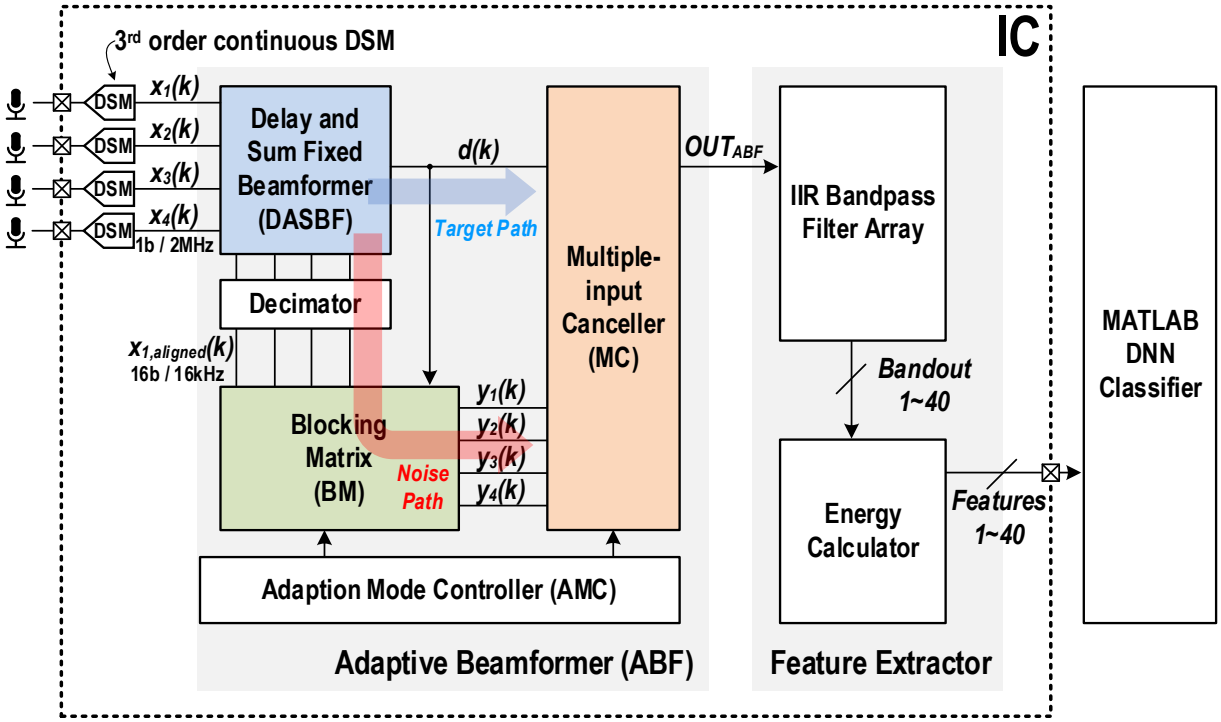


Figure 3-3. The prototype IC includes four DSM ADCs, an adaptive beamform processor, and a frequency-domain feature extractor.

Figure 3-3 shows a block diagram of the prototype system, including the RGSC ABF processor. Compared to frequency-domain sidelobe cancellers, the RGSC algorithm has three important advantages: (1) time-domain operation avoids the need for expensive FFT/IFFT and matrix calculations required in other ABF schemes [40]; (2) A time-domain algorithm tends to have less phase distortion; and (3) RGSC provides a flat directional response around the desired direction providing robustness to direction errors.

*Except for the ADC, all blocks are implemented in Verilog and synthesized.

Four Continuous-Time Delta-Sigma Modulator (CT DSM) ADCs digitize the analog signals from four microphones to generate $x_{1\text{-}4}(k)$. Bitstream-processing DASBF generates an initial estimate of the desired signal, d(k) from $x_{1\text{-}4}(k)$. Our ABF uses RGSC to remove noise from d(k). First, the blocking matrix (BM) adaptively removes the target signal from the time-aligned individual microphone outputs, $x_{1\text{-}4,aligned}(k)$, using the estimate, d(k), to produce noise components, $y_{1\text{-}4}(k)$. Next, a multiple-input-canceller (MC) adaptively subtracts $y_{1\text{-}4}(k)$ from d(k) to form the noise-reduced output, OUT_{ABF}. An adaptation-mode controller (AMC) selects whether to tune BM or MC coefficients depends on the noise condition. Finally, our system identifies frequency-domain energy features in the ABF output to facilitate speech recognition. Our time-domain beamformer does not use FFTs. We use simple bandpass filters to extract 40 features.

We directly process the bitstream outputs of the DSMs without decimation and downsampling to ensure sufficient beam-angle accuracy for RGSC. This greatly increases the effective sample rate and avoids the fundamental angular-resolution limit of conventional time-domain (i.e., delay-and-sum) beamforming. For example, an ADC sampling rate of 16kHz and a 2.54cm microphone spacing limit the angular resolution to only 60 degrees. Instead, our beamformer harnesses the relatively high sampling rate (2.048MS/s) of the DSMs for a much finer angular accuracy of 4.2 degrees.

3.2.2. Adaptive Beamformer

3.2.2.1 DASBF Operation

Figure 3-4 shows the detailed structure of the adaptive beamformer. First, the DASBF time-aligns and sums the single-bit outputs of the 4-channel 3rd order DSM ADCs with 3.4μs resolution. A 4th order cascaded integrator-comb (CIC) filter decimates the DASBF output (d_{2MHz}) and $x_{1\text{-}4,2MHz}$ by 128, producing d(k) and $x_{1\text{-}4,aligned}$. Decimation by 128 with an FIR filter requires many taps making it unsuitable for low power applications, while the CIC filter is efficient since it only uses adders and subtractors.

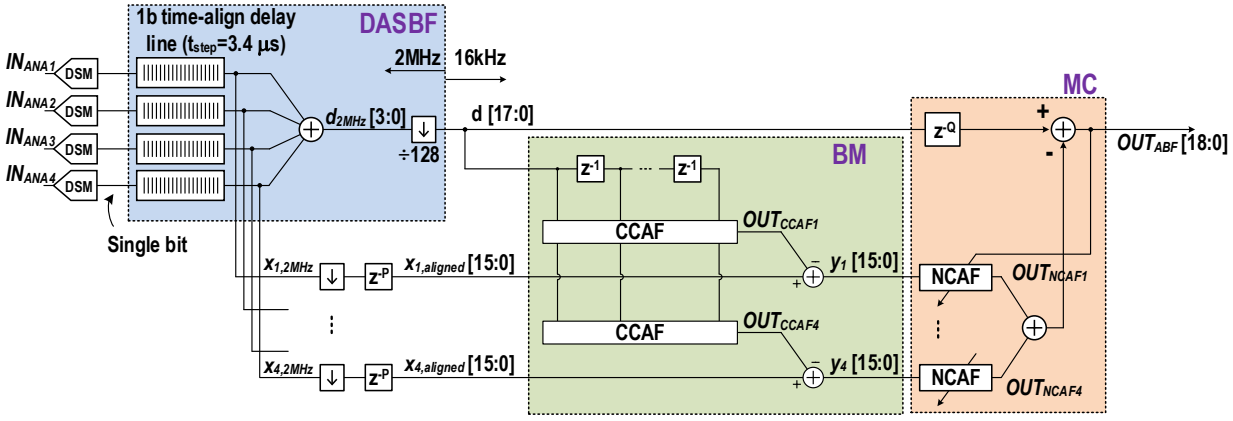


Figure 3-4. Detailed structure of the adaptive beamformer.

3.2.2.2 BM and MC Operation

A normalized-least-mean-squares (NLMS) algorithm in the blocking matrix (BM) adapts the coefficients of coefficient-constrained adaptive filters (CCAF) to minimize the desired signal component in the noise estimate, $y_{1\text{-}4}(k)$. The CCAF coefficients adapt quickly and accurately when the correlated signal between $x_{1\text{-}4,aligned}(k)$ and d(k) is strong. On the other hand, if the target signal is strong, the dominant component of $x_{1\text{-}4,aligned}(k)$ and d(k) is a target signal, hence the correlation is likely large. Thus, a strong target signal (i.e., high SNR) favors CCAF adaptation.

An NLMS algorithm in the MC adapts the norm-constrained adaptive filter (NCAF) coefficients to optimally subtract the noise estimate to form OUT_{ABF}. Noise is the main signal in the NCAF, indicating that NCAF adaptation favors high noise, low SNR situations. Hence, we separately enable adaptation for the BM and the MC depending on the estimated SNR. Therefore, the AMC (Figure 3-3) directs the BM to update the CCAF coefficients during periods of high SNR and allows the MC to update the NCAF coefficients during low-SNR periods [41]. The target signal and the noise are dominant components of d(k) and $y_{1\text{-}4}(k)$, respectively. Therefore, a simple shifter and lookup table roughly calculate the ratio between d(k) and $y_{1\text{-}4}(k)$ to estimate the SNR. Figure 3-5 shows its related equations.

$$y_m(k) = x_{m,aligned}(k-P) - H_m^T(k)D(k)$$
$$H(k) \triangleq \left[h_{m,0}(k), h_{m,1}(k), \dots, h_{m,N-1}(k)\right]^T$$
$$D(k) \triangleq [d(k), d(k-1), \dots, d(k-N+1)]^T$$
$$h'_{m,n} = h_{m,n}(k) + \alpha \frac{y_m(k)}{\|D(k)\|^2} d(k-n)$$
$$h_{m,n}(k+1) = \begin{cases} \phi_{m,n}, for\ h'_{m,n} > \phi_{m,n} \\ \varphi_{m,n}, for\ h'_{m,n} < \varphi_{m,n} \\ h'_{m,n}, otherwise \end{cases}$$
$$(m = 0, 1, \dots, M-1, n = 0, 1, \dots, N-1)$$

Coefficent-constrained adaptive filters (CCAF)

$$OUT_{ABF} = z(k) = d(k-Q) - \sum_{n=0}^{M-1} W_m^T(k) Y_m(k)$$
$$W_m(k) \triangleq \left[w_{m,0}(k), w_{m,1}(k), \dots, w_{m,L-1}(k)\right]^T$$
$$Y_m(k) \triangleq [y_m(k), y_m(k-1), \dots, y_m(k-L+1)]^T$$
$$W'_m = W_m(k) + \beta \frac{z(k)}{\sum_{j=0}^{M-1} \|Y_j(k)\|^2} Y_m(k)$$
$$W_m(k+1) = \begin{cases} \sqrt{\frac{K}{\Omega}} W'_m, for\ \Omega > K \\ W'_m, otherwise \end{cases}, \Omega = \sum_{m=0}^{M-1} \|W'_m\|^2$$

Norm-constrained adaptive filters (NCAF)

Figure 3-5. Equation of ABF operation [17].

We select 28 FIR filter taps as a compromise between BM performance and convergence time. Although using more taps leads to better filtering of the target signal, a larger filter increases convergence time and power consumption. For example, simulations show that if CCAF with a 28 tap FIR filter takes 0.2 seconds to converge, then a 40 tap filter takes 0.5 seconds.

The output of the adaptive beamformer, OUT_{ABF}, feeds to a log-Mel 40-band 4^{th} order IIR filter. A single shared multiplier calculates the power in each band. The frequency-bank energy features accumulate over 25ms [42]. Features overlap by 10ms and update at 67Hz.

3.2.2.3 Arithmetic Calculation

The proposed system optimizes the word length of the digital signal in each calculation stage depending on the precision requirements to achieve an energy-efficient implementation. We choose a fixed-point number system over a floating-point number system for two reasons. First, the calculation block is simpler because the fixed-point word length is smaller than with floating-point. Because the magnitude of the signal range is well-defined (e.g., a voltage-type digital signal is strictly limited to ±1), the extra exponent bit of floating-point is not necessary. Second, it is easier to implement since there is no need to handle the exponent part.

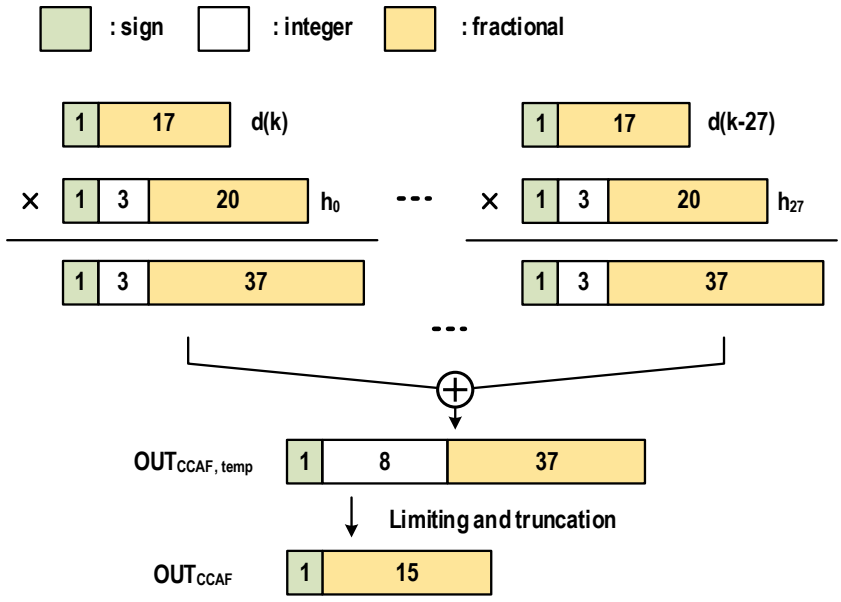


Figure 3-6. Calculation of fixed-point numbers in FIR filter. $h_{0\sim27}$ represents the coefficient of CCAF.

Figure 3-6 describes how the CCAF FIR filter calculates with fixed-point. We chose a 28-taps FIR filter considering the following trade-off: large taps give better performance of BM since it can cover lower frequency, but it takes a longer time to converge and consumes more power. The d(k) integer bit is zero since the analog input is always less than 1V. The fractional bit of CCAF coefficients ($h_{0\text{-}27}$) is 20 for accurate convergence. We set the integer part of $h_{0\text{-}27}$ as 3 bit since the simulations show the coefficients do not exceed 8 with enough margin. The product of d(k) and $h_{0\text{-}27}$ has a fractional bit as 37 (=17+20), which is a sum of fractional bits of d(k) and $h_{0\text{-}27}$ to preserve the precision before addition. Next, we need 5 extra integer bits when adding 28 taps to cover the maximum range (ceiling of $\log_2 28$). Meanwhile, the range of $OUT_{CCAF,temp}$ is also 1V since it tracks the input desired signal. Hence, the system reduces word length for hardware efficiency by removing the integer part of $OUT_{CCAF,temp}$ through limiting and truncating the fraction part to 15 to meet the required precision.

3.2.2.4 Hardware Sharing

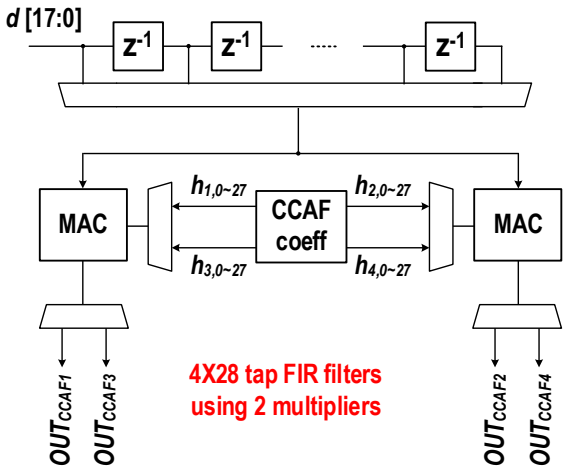


Figure 3-7. CCAF filter implementation with hardware sharing.

We share blocks and simplify some operations to reduce leakage power and die area. Operating at the decimated data rate of 16kHz would result in an unacceptably large die area and leakage power. We share four multipliers through multi-phase operation to implement the four channels of 2 x 28 tap FIR filters for CCAFs and NCAFs, as shown in Figure 3-7. We also share arithmetic blocks used in coefficient calculation. The shared blocks operate at the 2.048MHz DSM clock rate. Sharing multipliers in the BM block reduces area by 36x, and sharing multipliers and dividers in the MC reduces area by 40x. Approximation with a lookup table further reduces the area and power consumption of the square-root operations in the NCAF coefficient calculations.

Figure 3-8 shows the timing diagram of internal signals regarding hardware sharing. First, DASBF generates d(k) by summing time-aligned $x_{1\text{-}4,2MHz}(k)$ at t_1. At this moment, $CCAF_{coeff,1\text{-}4}(k)$ is ready, while $NCAF_{coeff,1\text{-}4}(k)$ is not. Then BM starts calculating $y_{1\text{-}2}(k)$ using $CCAF_{coeff,1\text{-}2}(k)$ at t_1, and finishes at t_2. Since $y_{1\text{-}2}(k)$ and $y_{3\text{-}4}(k)$ calculations share the same hardware, the calculation of $y_{3\text{-}4}(k)$ can start at t_1 after $y_{1\text{-}2}(k)$. At t_1, BM can calculate $CCAF_{coeff,1\text{-}2}(k)$ since $y_{1\text{-}}$

$_2$(k) are ready. At t_2, $y_{3\text{-}4}$(k) are ready, so BM starts calculating $CCAF_{coeff,3\text{-}4}$(k). Meanwhile, at t_2, $y_{1\text{-}4}$(k) is ready. So, MC starts calculating all $OUT_{NCAF1\text{-}4}$(k) at the same time. Since it requires more calculation time than $y_{1\text{-}4}$(k), each channel has its own calculation hardware and calculates simultaneously. Then at t_4, $OUT_{NCAF1\text{-}4}$(k) is ready, and the final output OUT_{ABF}(k) comes out. Finally, MC calculates $NCAF_{coeff,1\text{-}4}$(k) simultaneously from t_4 using four calculation hardware and finishes it when needed.

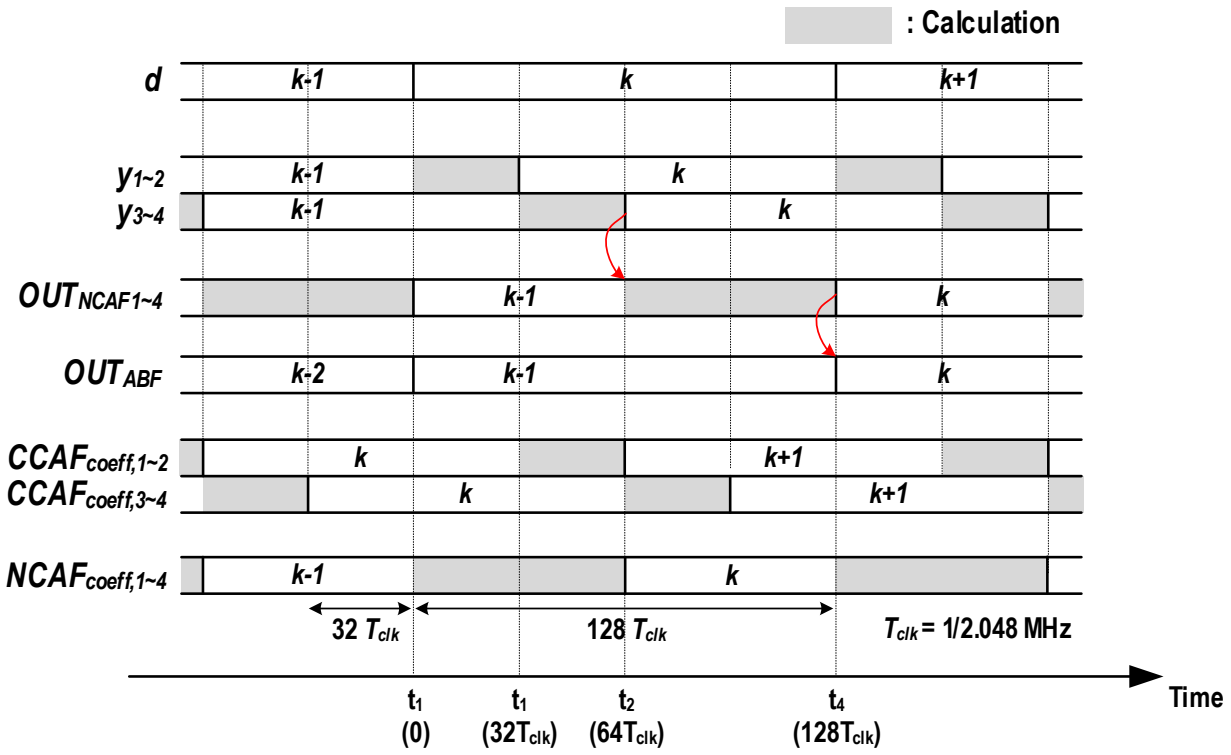


Figure 3-8. Timing diagram of signals and coefficients calculation.

3.2.2.5 Clock Frequency Optimization

We optimize the DSP clock speed for energy efficiency. Decimation enables much slower processing in blocks that do not need high time accuracy. The slower processing rate reduces dynamic power and facilitates hardware sharing, reducing both die area and leakage power. For instance, a single 28-tap FIR filter operating at 16kHz is 90 times more power-efficient than one running at 2.048MHz (Figure 3-9). While the multipliers and adders are simpler for 2.048MHz operation due to smaller bit-widths, each arithmetic operation occurs 128 times more often, resulting in higher overall power consumption. Also, the 2.048MHz FIR filter occupies 15 times more area than the 16kHz one. Furthermore, hardware sharing in a 2.048MHz filter does not help much because the non-sharable delay line occupies most of the area. Also, sharing multipliers for 2.048MHz operation require a higher main-clock frequency.

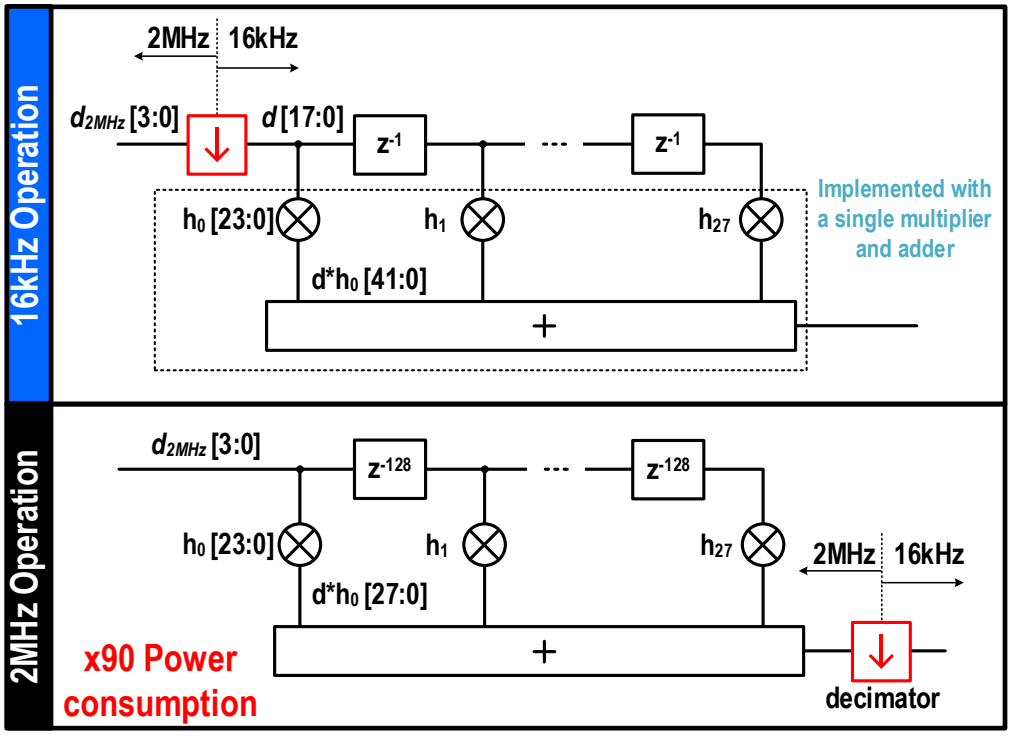


Figure 3-9. Two ways to implement an FIR filter after (16kHz) and before decimator (2MHz) with identical functionality. The 16kHz case consumes 90 times less power.

3.2.2.6 CIC Filter (decimator) Implementation

We use the MATLAB [43] Filter Designer tool to implement the decimator as shown in Figure 3-10. We choose the CIC filter as a decimator due to its simplicity. We choose the 'number of sections' to be 4 to suppress shaped quantization noise since the ADC is third-order noise-shaping [44]. The output word length should support the target SNR of the signal. We use the MATLAB 'Generate HDL' function and make Verilog code. We set the parameters for 4-bit signed input d(k) as shown in Figure 3-11. After auto-generating Verilog code, we modify the output bit assignment as shown at the bottom of Figure 3-11 by matching the input and output range through simulation.

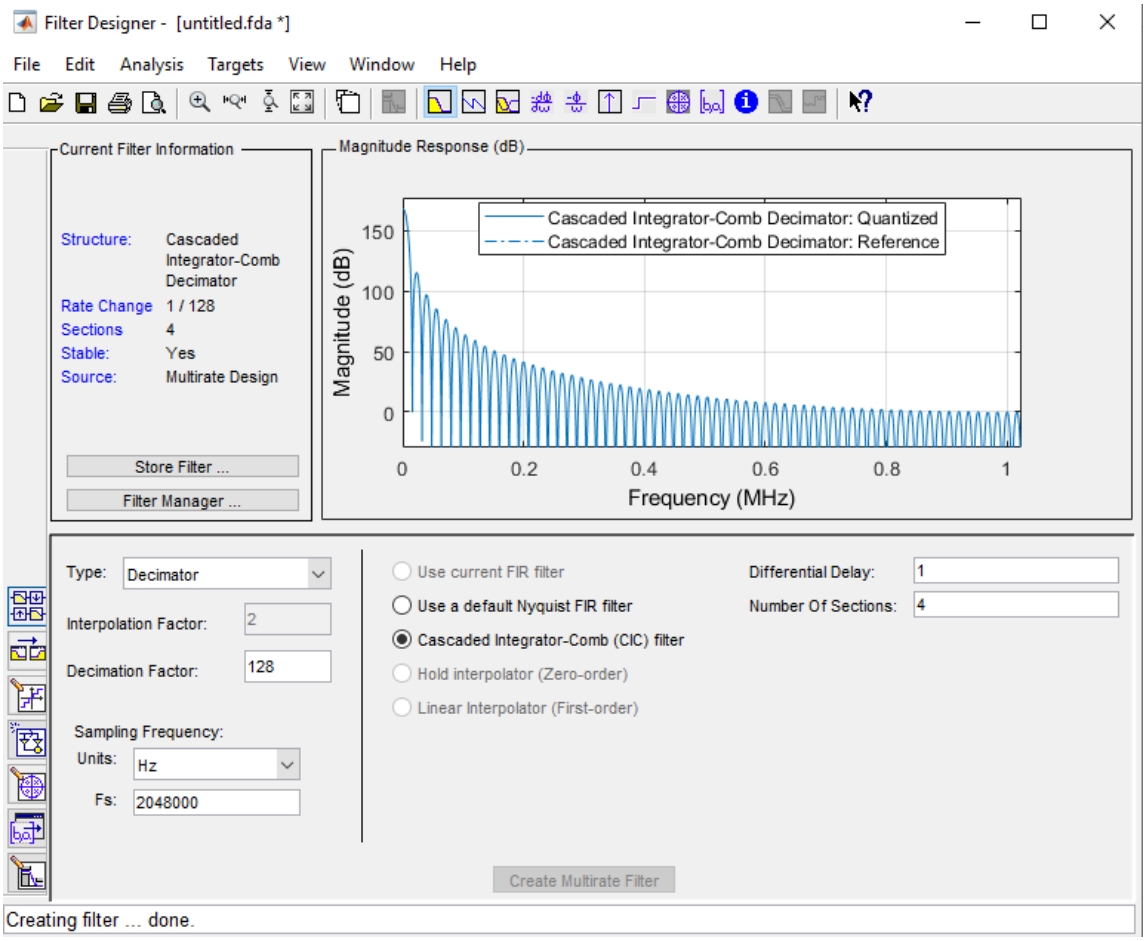


Figure 3-10. CIC decimator implementation with the MATLAB filter designer.

```
// -------------------------------------------------------------
// HDL Implementation    : Fully parallel
// -------------------------------------------------------------
// Filter Settings:
//
// Discrete-Time FIR Multirate Filter (real)
// -----------------------------------------
// Filter Structure       : Cascaded Integrator-Comb Decimator
// Decimation Factor      : 128
// Differential Delay     : 1
// Number of Sections     : 4
// Stable                 : Yes
// Linear Phase           : Yes (Type 1)
//
// Input                  : s4,1
// Output                 : s18,-13
// Filter Internals       : Minimum Word Lengths
//   Integrator Section 1  : s32,1
//   Integrator Section 2  : s32,1
//   Integrator Section 3  : s31,0
//   Integrator Section 4  : s26,-5
//   Comb Section 1        : s23,-8
//   Comb Section 2        : s22,-9
//   Comb Section 3        : s21,-10
//   Comb Section 4        : s20,-11
// -------------------------------------------------------------

// Manually moidified part by author
assign output_typeconvert = section_out8[18:1];
```

Figure 3-11. MATLAB filter design parameters for 4-bit signed input decimator.

3.2.3. Continuous-time Delta-sigma Modulator

Figure 3-12 shows the 3rd order CT DSM [12] used in the proposed system. A CT DSM has several advantages over a discrete-time (DT) DSM. First, a CT DSM has an inherent anti-aliasing filter [45], removing the need for an additional input low pass filter. Second, a CT DSM has a resistive input. Hence the preamplifier does not need to drive the large sampling capacitor of DT DSM, so the overall system efficiency, including the microphone, can be improved. Third, a CT DSM relaxes the performance requirements of 1st stage amplifier. Because CT DSM does not have a settling operation, the required amplifier gain bandwidth is much smaller than a DT DSM.

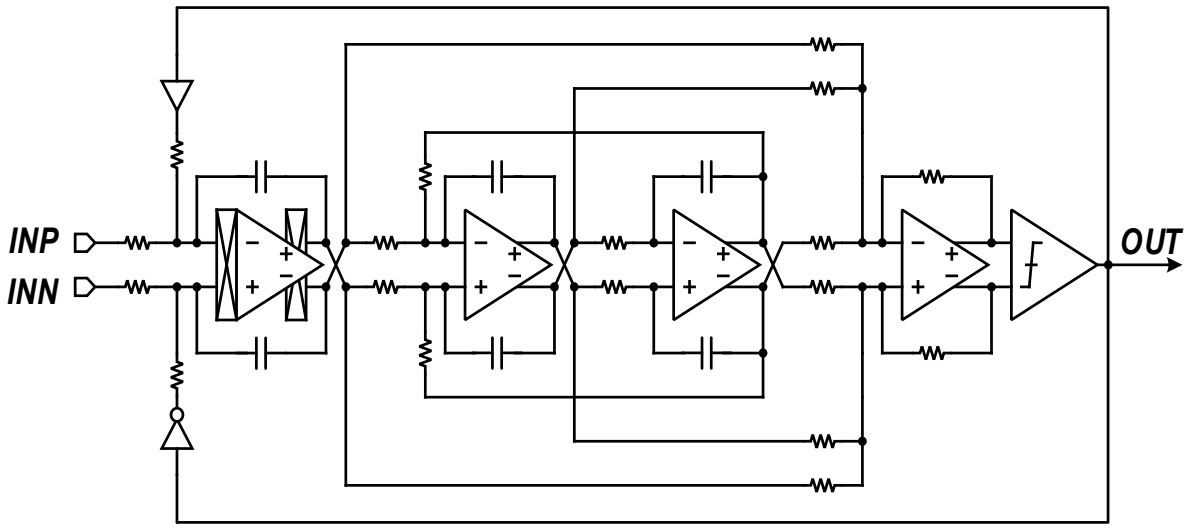


Figure 3-12. Schematic of 3rd order continuous-time delta-sigma modulator.

* Seungjong Lee is the primary designer of ADC.

3.3. Measurements

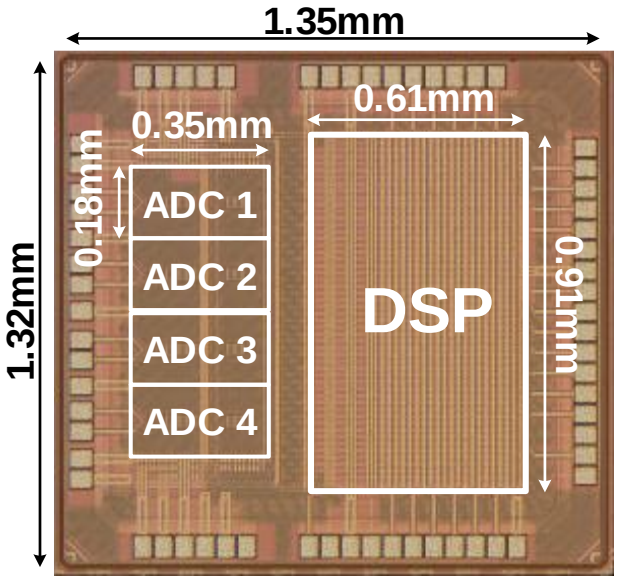


Figure 3-13. Die micrograph.

The prototype is fabricated in 40nm LP CMOS and has an active area of 0.89mm^2 (Figure 3-13). The measured SNDR of the ADC is 83.3 dBA. Note that dBA is A-weighted decibels and is based on the response of the human ear. This weighting is implemented in MATLAB [46] and modifies the FFT spectrum when calculating SNDR.

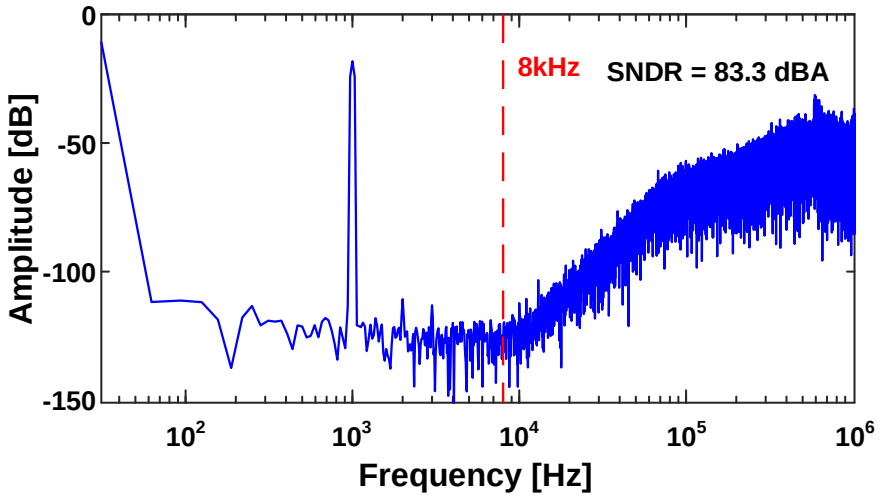


Figure 3-14. Measured 32k point FFT for a single CT DSM.

3.3.1. Test Setup

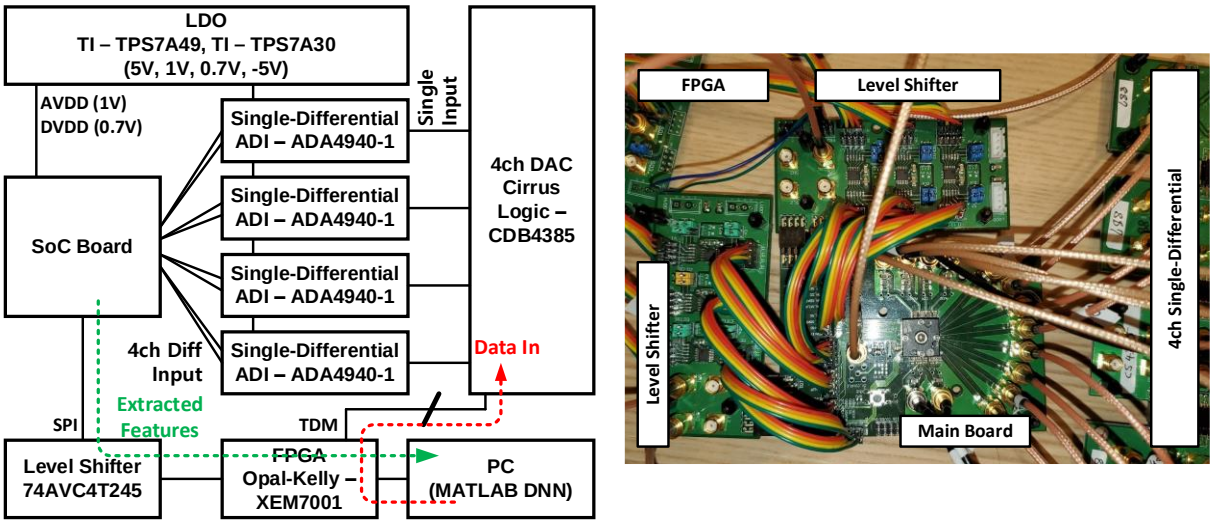


Figure 3-15. Board diagram and photo of the test setup.

A 4-channel audio DAC (Cirrus Logic CDB4385) controlled by an FPGA applies audio inputs to the ADCs to emulate microphone inputs. The sound signal is far-field and anechoic. External single-to-differential amplifiers (Analog Devices ADA4940) convert the single-ended DAC outputs to the differential with a 0.5V bias.

3.3.2. Coefficient Adaptation

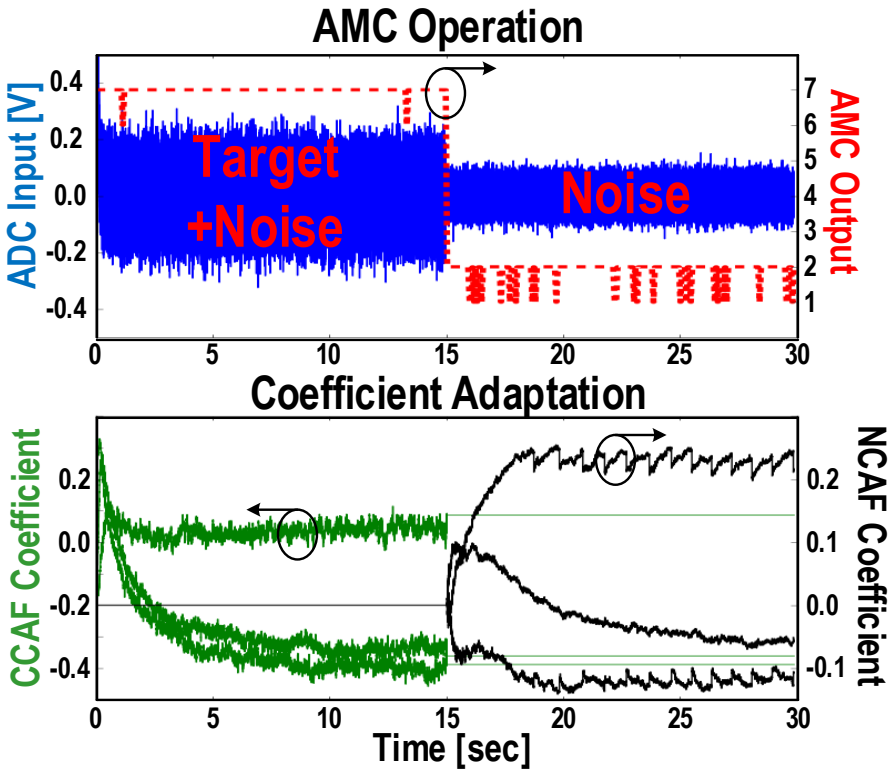


Figure 3-16. AMC controls the adaptation mode by estimating SNR (upper), and its coefficients convergence (bottom).

Figure 3-16 tracks the output of the AMC and the coefficient update. While there is a signal (i.e., 0~15s), the AMC output is high, indicating a strong target signal in the given direction. Hence, the BM starts to adapt the CCAF coefficients. On the other hand, when there is no target signal (15~30s), the AMC output becomes low, and the MC adapts NCAF coefficients. After the coefficients convergence, we fix the coefficients and measure the beampattern.

3.3.3. Measured Beampattern

Figure 3-17 shows simulations of the noise suppression for different DOAs of noise. We use a cardioid microphone array, as shown in Figure 3-18, and fix the target DOA while sweeping the noise DOA. For each noise DOA, we adapt the beamformers coefficients with 15dB SNR. The beamformer consistently suppresses noise over a wide range of noise DOA regardless of the target DOA. However, if the noise DOA is within 10 degrees of the target DOA, the beamformer cannot fully suppress it.

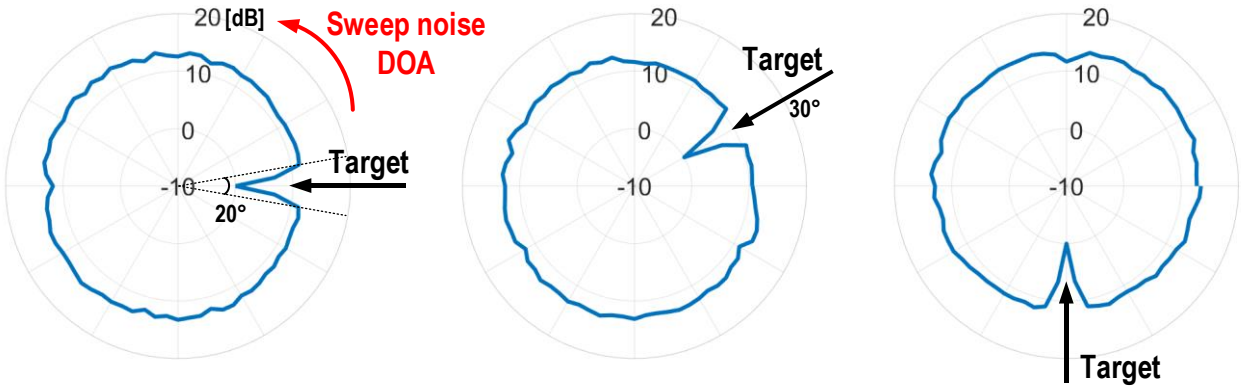


Figure 3-17. Simulated noise rejection for a sweep of the DOA of Gaussian noise.

Figure 3-18 plots measured beam patterns for different noise environments. It is clear from the measurements that the beamformer adaptively places nulls in the noise direction while maintaining a near-unity gain in the desired direction. Furthermore, we notice the adaptive beamformer assigns a larger-than-unity gain to the noise-free direction - this does not affect performance as the beamformer adapts its null if the noise-source direction changes (e.g., to 270°, Figure 3-18), then the beamformer correctly adapts its null to that angle.

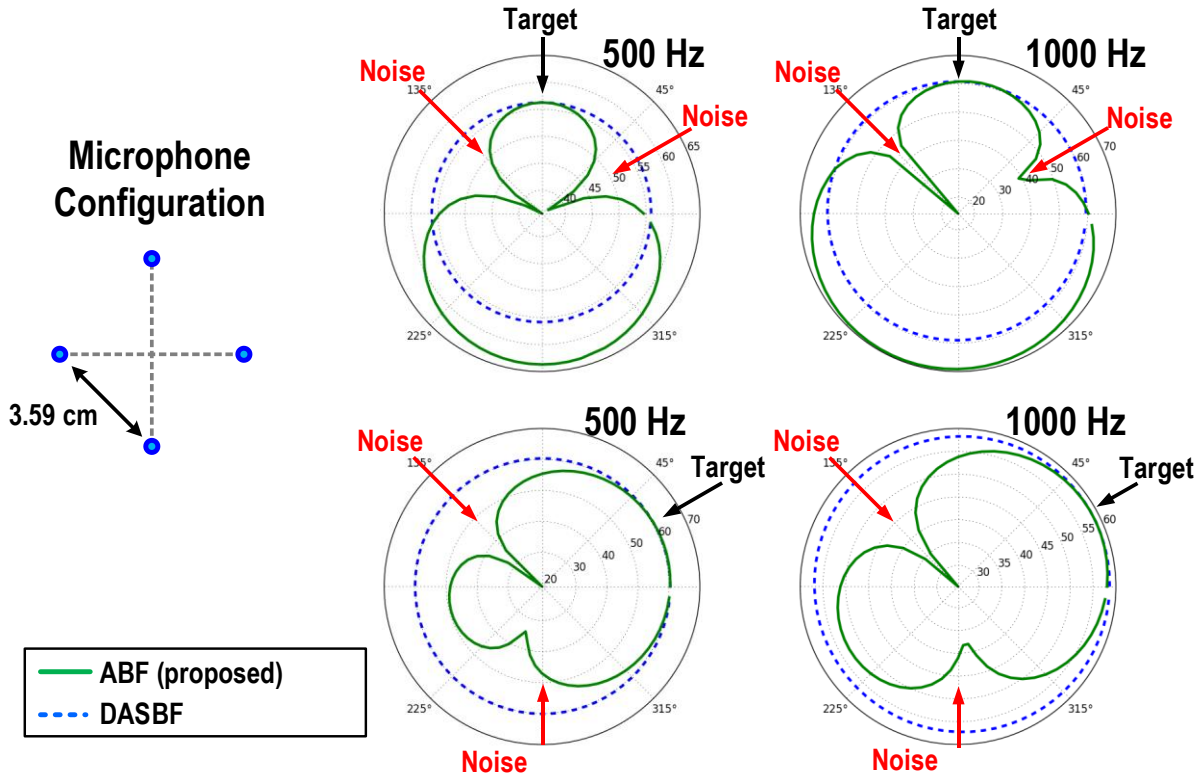


Figure 3-18. Cardioid microphone configuration (left), measured beamforming patterns for adaptive beamforming (ABF) and fixed DASFB with different noise directions (right). ABF automatically directs the nulls towards the noise sources.

3.3.4. Speech Recognition Test

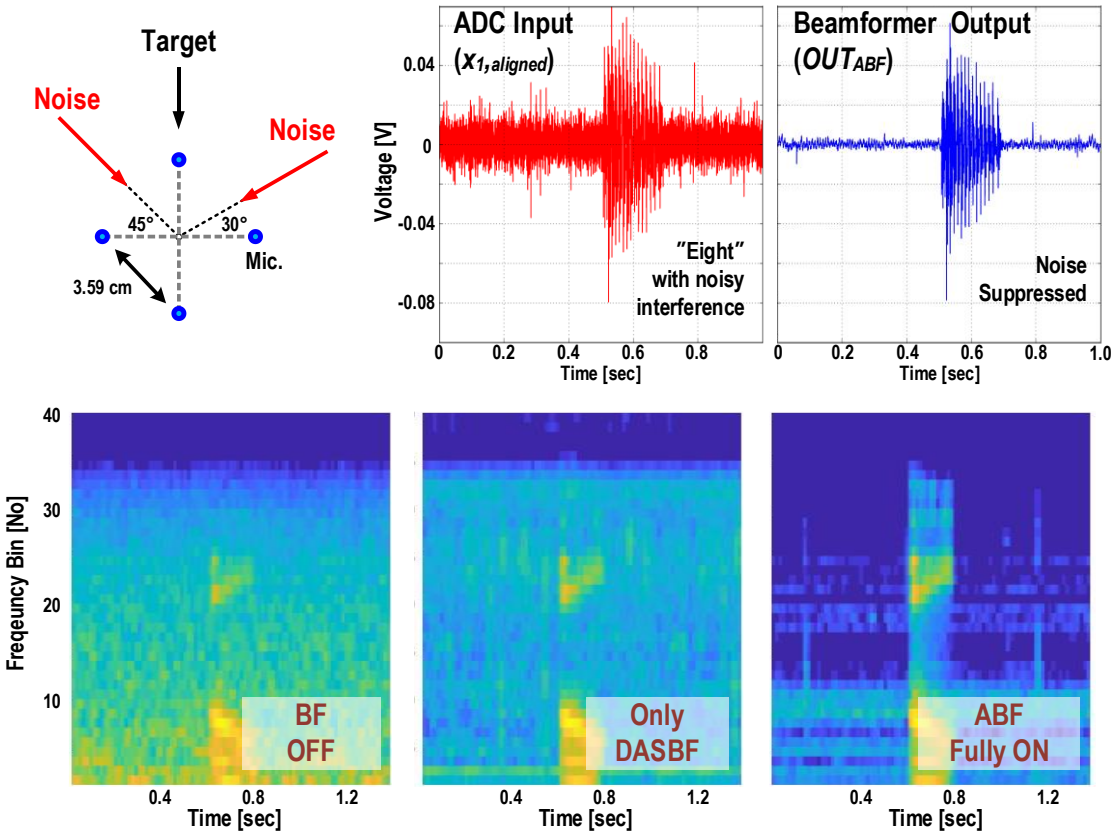


Figure 3-19. (top) Signal and noise directions, beamformer input and output and (bottom) Mel features generated by chip without beamforming, with DAS beamforming and with adaptive beamforming.

We use the Tensorflow keyword dataset [39] to demonstrate the noise suppression of adaptive beamforming. In this test, a 4-channel 24-bit DAC array replaces the microphones. The FPGA collects a 1-second duration of the 40 features generated by the chip, transmits this data to a PC, which runs MATLAB DNN [47] for classification. The dataset consists of 1200 utterances of 9 words, 2000 unknown sounds, and 2000 background noise samples. The dataset is divided into training and validation with an 8 to 1 ratio. The measured recognition accuracy without noise is 93.5%. To assess the advantages of adaptive beamforming, we measure recognition accuracy with two noise sources after adaptation. The target signal direction is at 90°, while the two noises

are placed at 30° and 135°. The noise is random background noise from Tensorflow and is 15dB lower in power than the target signal. Figure 3-19 shows that ABF dramatically improves the spectrogram. The adaptive beamformer improves speech recognition accuracy from 64% (no beamforming) to 90%.

Figure 3-20 shows the simulated recognition accuracy for each word. Single-syllable words are more affected by noise because the high-frequency noise can easily obscure consonants, which are usually high-frequency and small in magnitude. The beamformer effectively suppresses high-frequency noise (Figure 3-18) and hence significantly improves accuracy. A very low SNR may cause misdetection by the AMC, leading to inappropriate adaption. Simulations with incorrect AMC operation show a 10-15% degradation in recognition accuracy.

Accuracy [%]	down	eight	go	happy	no
w/o BF	72	68	59	76	49
with ABF	91	93	89	93	87
	seven	stop	up	yes	unknown
w/o BF	66	71	49	83	71
with ABF	89	91	92	94	85

Figure 3-20. Accuracy for 9 words and unknown.

3.3.5. Power Consumption Analysis

The total measured power consumption is 0.65mW from 1V analog and 0.7V digital supplies. Figure 3-21 shows a breakdown of the power consumption. A single ADC consumes 91μW, and half of this power is consumed by a first-stage amplifier. The BM and MC blocks are responsible for 70% of the digital power consumption because the coefficient calculations include multiplication and division. Table 1 compares ASR frontends. Only our system integrates both high-resolution ADCs and adaptive beamforming and does so with sub-mW power consumption.

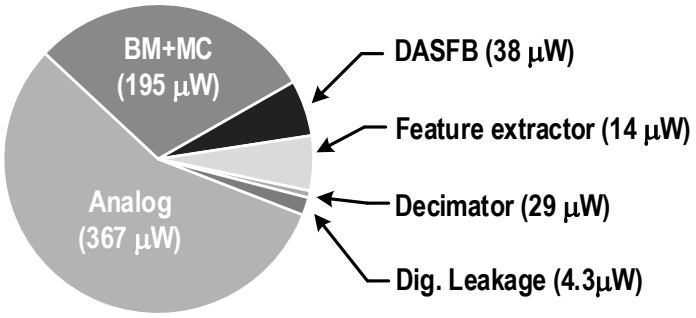


Figure 3-21. Power consumption breakdown.

Table 1: Comparison with high-SNDR beamforming feature extraction systems

	This Work	**[12] Lee**	**[48] Liu**	**[27] Sainath**	**Google Home**	**Amazon Echo**
Implementation	**Analog mic. +Single chip**	Analog mic. +Single chip	Digital mic. +Multichip	Digital mic. +Software	Digital mic. +Multichip	Analog mic. +Multichip
Technology	**40nm LP CMOS**	40nm GP CMOS	90nm CMOS	-	-	-
Area (mm^2)	**0.89**	1.1	0.47	N/A	N/A	N/A
VDD (Analog / Digital)	**1.0V / 0.7V**	1.0V / 0.55V	- / 0.33V	-	-	
# Signal Sources	**4**	8	2	2	2	7
Functionality	**ADCs, beamforming, feature extraction**	ADCs, beamforming, feature extraction	Beamforming (no steering), feature extraction	Beamforming, feature extraction, classification	ADCs, beamforming, feature extraction, classification	ADCs, beamforming, feature extraction, classification
DR [8kHz BW]	**83dBA**	85dBA	-	-	108dBA	98dBA (mic.) 97dBA (ADC)
BW	**8kHz**	8kHz	8kHz	8kHz	8kHz	8kHz
Beamforming Type	**Adaptive RGSC**	Fixed delay-and-sum	Adaptive Griffiths-Jim	Adaptive filter-and-sum with trained coefficients	-	-
Feature Type	**Log-Mel filter bank energy**	Mel filter bank energy	FFT-based Log filter bank	Convolutional long short-term memory DNN filter bank	-	-
# Features	**40**	60	8	128	-	-
Power Consumption	**0.65mW**	3.95mW	0.1mW*	-	4.4mW**	47mW**

* Excludes ADCs, ** Calculated from datasheets, only includes MEMS microphones, ADCs.

Chapter 4. A Multi-Mode Speech Recognition Frontend with Self-DOA Correction Adaptive Beamformer

4.1. Motivation

Fixed delay-and-sum (DAS) beamforming in Chapter 2 is simple to implement, but only suppresses noise from a fixed direction of arrival (DOA) [12]; hence, it is ineffective in real varying noise conditions. [49] implements ultra-low-power keyword spotting (KWS) with noise suppression, but the lack of an ADC and beamforming limit practical application.

In Chapter 3, we focus on adaptive beamforming (ABF), which actively adjust nulls to suppress varying noise sources. Adaptive beamforming with a trained DNN is promising [21]-[31] but requires extensive training data and high power consumption and is not applicable for battery-operated systems. The prototype in Chapter 3 [18] (Figure 4-1) adaptively reduces noise and interference in the output of a fixed DAS beamformer with reasonable power consumption and an accurate steering angle. However, the prototype is still hampered by: 1) High DSP power consumption due to high ADC sampling rate and the need for complex calculations, especially in the blocking matrix (BM); 2) Target signal direction errors in DAS cause severe signal distortion; and 3) Worst-case input-SNR design causes high ADC and DSP power regardless of actual signal conditions.

*This work was done in collaboration with Seungjong Lee and Seungheun Song. They are the main contributors to ADC design.
Some text and figures are from [32].

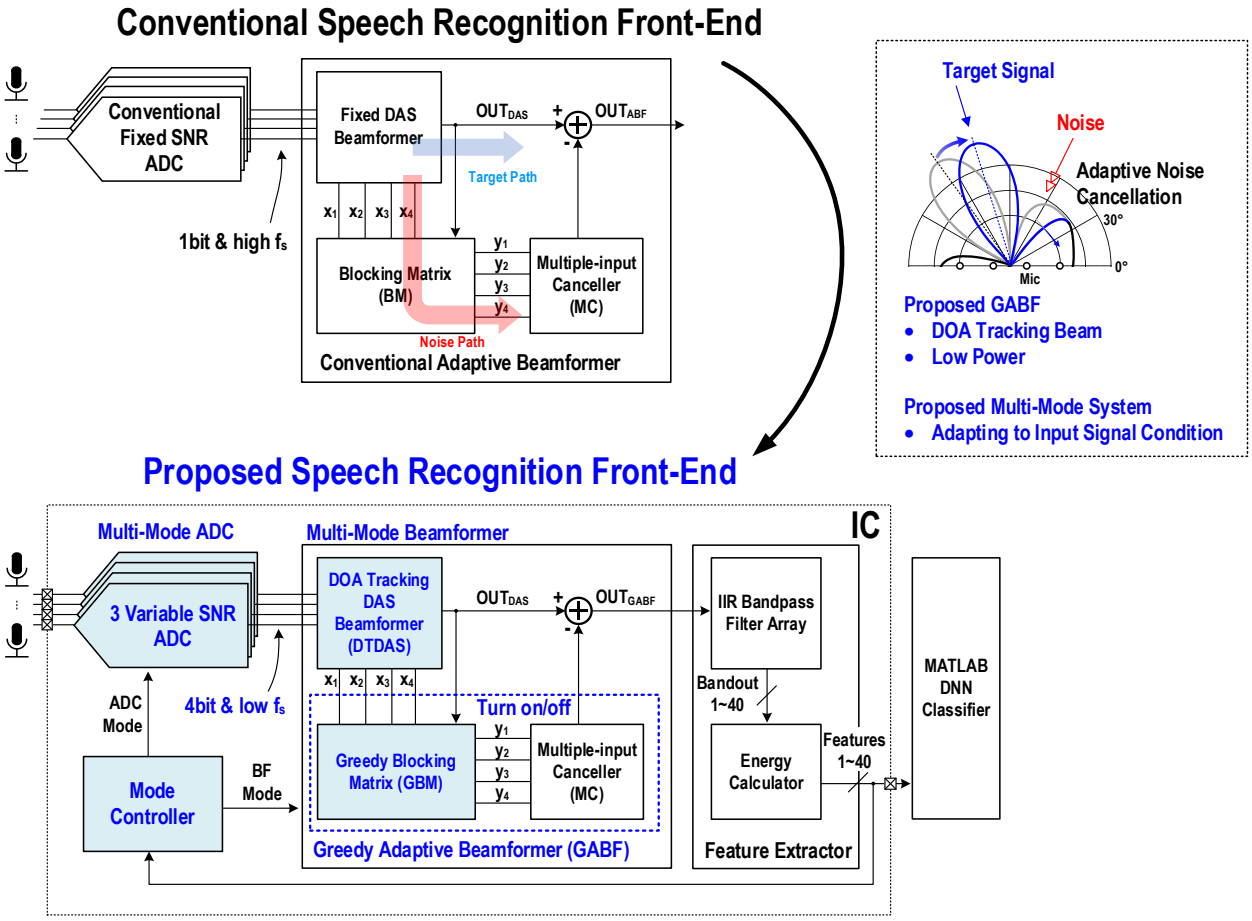


Figure 4-1. Conventional Adaptive Beamformer [18] (top) and proposed multi-mode automatic speech frontend end with Greedy Adaptive Beamformer (GABF) and multi-mode ADCs (bottom).

Our ASR frontend system (Figure 4-1) addresses the problems of the prototype ABF in Chapter 3 with: (1) Low DSP power consumption (3x lower than state-of-art ABF [18]) thanks to an innovative greedy blocking matrix (GBM) with simple calculations and a reduced-sample-rate ADC; (2) Automatic DOA error compensation with a DOA tracking DAS beamformer (DTDAS) aided by the GBM, (3) A multi-mode hybrid ADC architecture adapts to signal conditions and consumes an order-of-magnitude less power than the state-of-art; and (4) Multi-mode beamforming takes advantage of high signal SNR to reduce total power consumption by up to 46%.

4.2. System Implementation

4.2.1. System Overview

Our speech-processing frontend connects to four microphones and outputs frequency-domain speech features (Figure 4-1). After digitization by the multi-mode ADC array, the DTDAS beamformer corrects speaker-direction estimate errors by appropriately time-aligning the ADC outputs. First, DTDAS generates an enhanced target signal OUT_{DAS} and outputs correctly aligned ADC outputs, $x_{1\text{-}4}$. Next, the GBM removes the target signal from $x_{1\text{-}4}$ to generate a residual noise-dominant signal, $y_{1\text{-}4}$. Then, an FIR-based multiple-input canceller (MC) cancels the noise in OUT_{DAS} to produce a clean beamformer output, OUT_{GABF}. Finally, a feature extractor generates 40 log-Mel frequency energy features [50] for speech recognition. A mode controller estimates target power and noise power floor [51] from the output of the feature extractor. The mode controller controls greedy adaptive beamformer (GABF) coefficient adaptation, beamformer mode, and ADC mode.

*Except for the ADC, all other blocks are implemented in Verilog and synthesized.

4.2.2. Greedy Adaptive Beamformer (GABF)

4.2.2.1 Greedy Blocking Matrix (GBM)

Figure 4-2 shows a conceptual diagram of the blocking matrix (BM). The primary role of BM is to generate a noise-dominant signal, y_m, from the inputs. The simplest BM is a fixed BM [20] shown in Figure 4-2 (bottom-left) using only a subtractor. If x_m and x_{m+1} contain the same target signal and their phases are aligned, a simple subtraction can remove the common target signal and result in noise-dominant, y_m. However, a DOA error (i.e., x_m and x_{m+1} are not aligned) causes leakage of the target signal into y_m, which is not desirable. To compensate for the alignment error, we add a variable delay cell to align the signals.

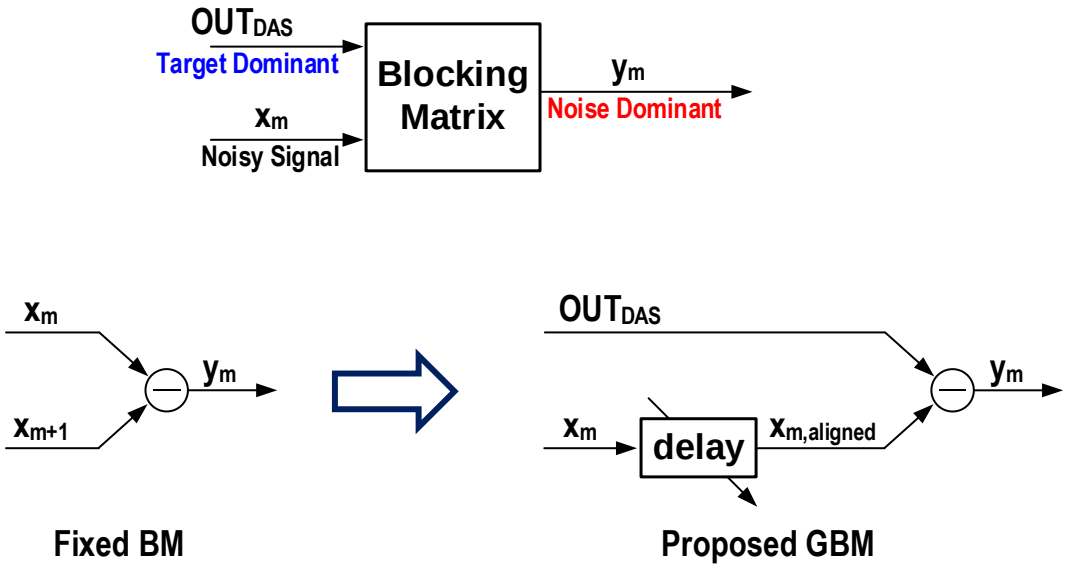


Figure 4-2. Conceptual diagram of GBM.

Figure 4-3 shows how the adjustable delay cell reduces target signal leakage in y_m. For ease of understanding, we choose single-tone sinewave as a target signal. Figure 4-2 (left) shows that non-alignement between d and x_m cause leakage of the target signal (sinewave) into y_m. This

leakage causes signal distortion in the next block, the MC. On the other hand, the variable delay cell aligns d and $x_{m,aligned}$ (right waveform in Figure 4-3), and then subtraction results in y_m without target signal leakage.

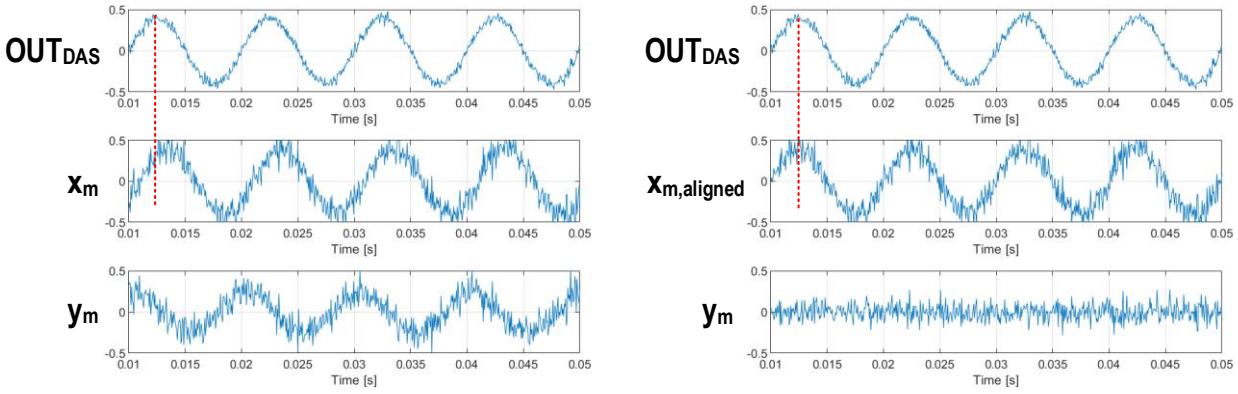


Figure 4-3. Example of GBM waveforms.

We implemented an adaptive beamformer with a newly proposed blocking matrix (Figure 4-4) using variable delay lines ($TD_{1\text{-}4}$ and $TDC_{1\text{-}4,a\text{-}c}$). DOA errors are inevitable due to speaker movement or DOA estimation errors from external sources. The timing misalignment causes significant signal distortion, especially for high-frequency signals due to their short wavelength. This is critical for speech because high frequencies (i.e., consonants) are vital for speech intelligibility. However, a conventional BM cannot correct direction errors since it uses fixed DAS [18]. Furthermore, the FIR filters in a conventional BM cause high power (80x more than the proposed GBM) and slow adaptation (e.g., 5s [18]). Instead, the GBM generates a noise-dominant signal with simple adjustable delay lines ($TD_{1\text{-}4}$) and automatically corrects DOA errors.

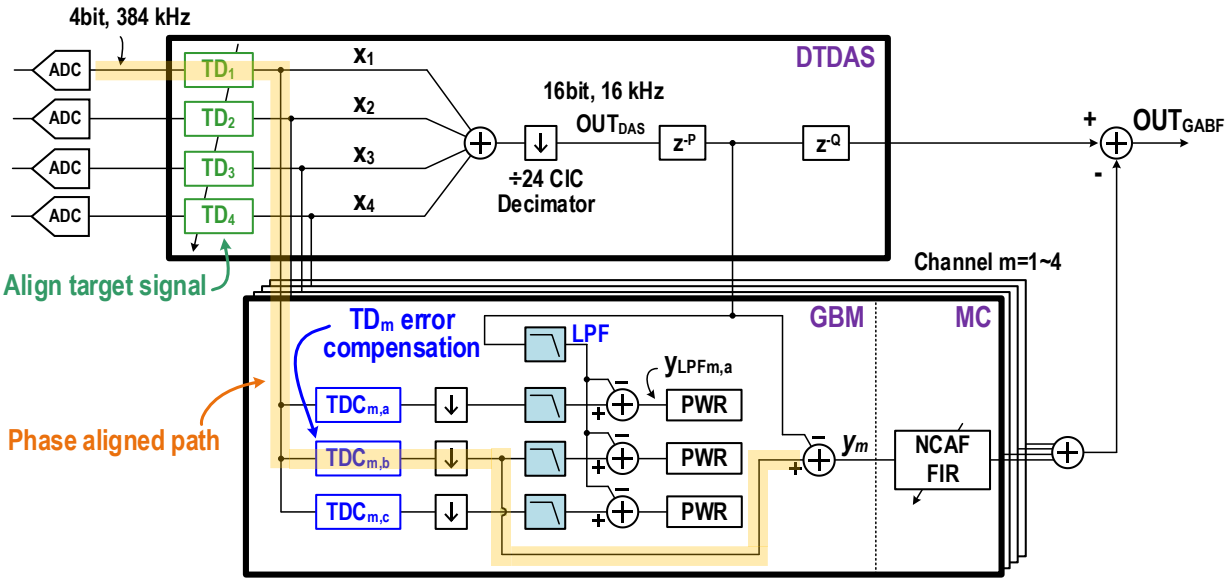


Figure 4-4. Block diagram of the proposed GABF.

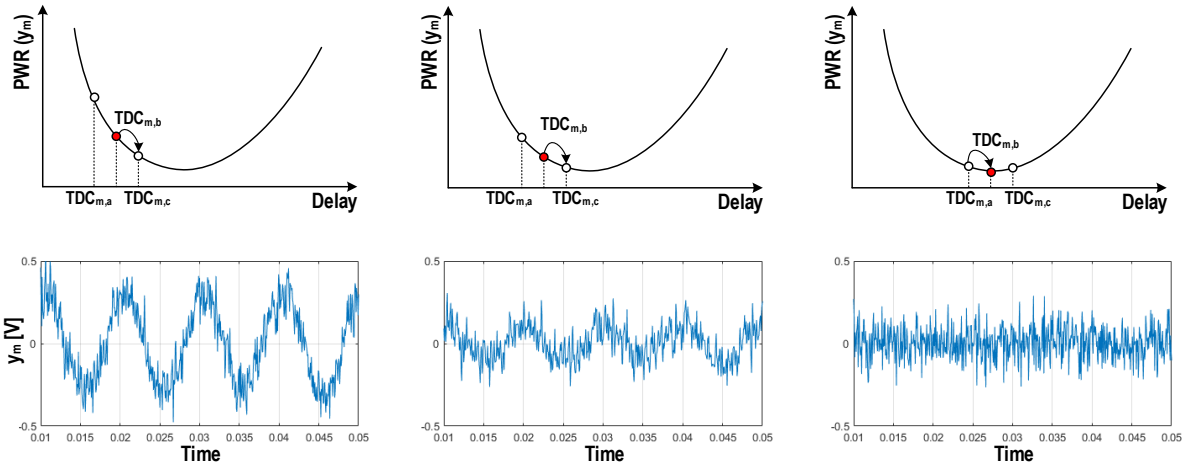


Figure 4-5. Greedy algorithm to find optimum time delay.

A greedy algorithm adapts the delays (TD_m and TDC_m), as shown in Figure 4-5. For each channel, three trial delay compensators ($TDC_{m,a\text{-}c}$) assist in DOA correction. When OUT_{DAS} and $x_{1\text{-}4}$ correctly align, subtracting OUT_{DAS} from $x_{1\text{-}4}$ results in a noise-dominant signal, $y_{1\text{-}4}$. Therefore, we simultaneously apply three different $TDC_{m,a\text{-}c}$, and calculate the y_m average power for each case to find the optimum delay. We assume that the target signal has a larger power than

noise. Hence, the GBM selects the $TDC_{m,a\text{-}c}$ path with the minimum y_m power, indicating the best removal of the target signal. Next, the GBM updates the $TDC_{m,a\text{-}c}$ delays with the previously chosen TDC_m as the center ($TDC_{m,b}$) delay, and repeats the process. As a result, $TDC_{m,b}$ gradually approaches the optimum value (Figure 4-5).

We can correct DOA errors in $TD_{1\text{-}4}$ by observing $TDC_{1\text{-}4,b}$. When DTDAS accurately steers toward the target signal, $TD_{1\text{-}4}$ of DTDAS align the $x_{1\text{-}4}$ perfectly; hence the compensation delay $TDC_{m,b}$ in the 'phase aligned path' (Figure 4-4) should do nothing (i.e., $TDC_{m,b} = 0$). In other words, if TDC_m is non-zero, it indicates that errors exist in $TD_{1\text{-}4}$. Hence, the proposed GBM feeds a partial $TDC_{m,b}$ (e.g., $TDC_{m,b}/4$) to TD_m to compensate TD_m's error (Figure 4-6). After some iterations, $TD_{1\text{-}4}$ is optimally tuned, and $TDC_{m,b}$ reaches zero, indicating that DOA estimation error is corrected.

The advantage of DOA correction with GBM is that it does not require the knowledge of prior microphone locations. Also, GBM keeps the simple calculation even with the number of microphones increasing. We describe the future work related to this advantage in Chapter 4.4.

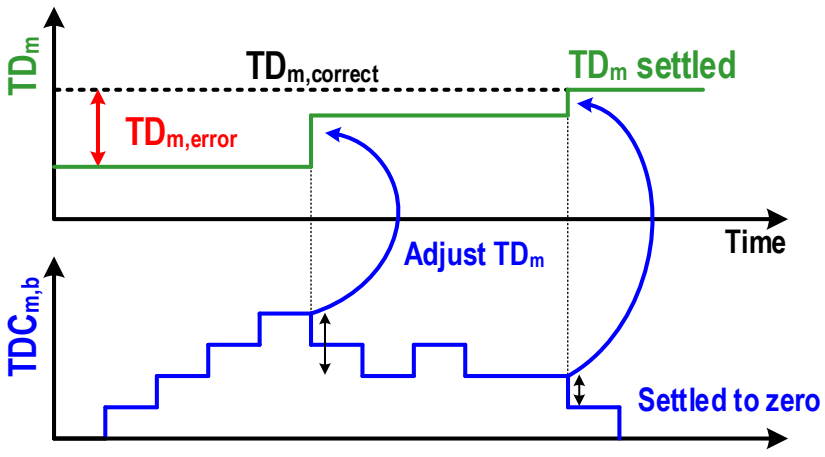


Figure 4-6. TD_m update for DOA correction by GBM.

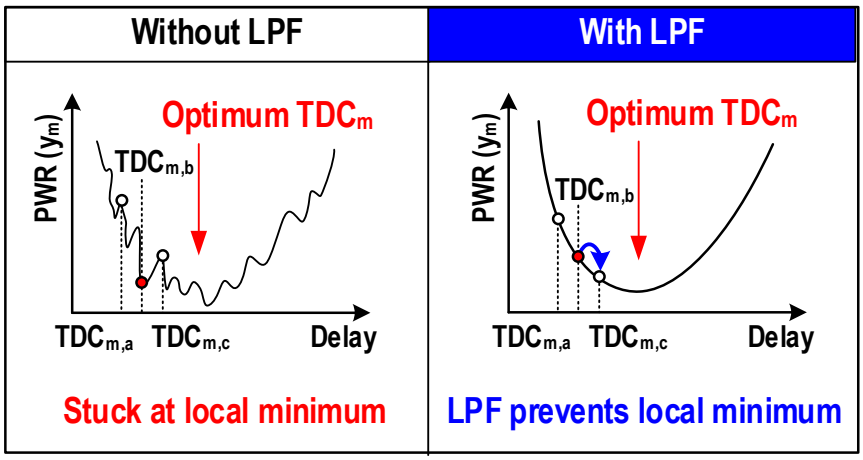


Figure 4-7. Greedy algorithm to find optimum time delay.

A potential problem with the greedy algorithm is local minima. For example, if the space between microphones is longer than half of the wavelength, multiple optimum points are causing spatial aliasing. Hence, the GBM uses low-pass filters (LPF) to prevent spatial aliasing (Figure 4-7). These filters do not degrade the search performance since most speech energy lies at low frequency. We design a "Direct-Form FIR" low-pass filter using the MATLAB filter design tool [52] -Figure 4-8 shows the MATLAB code and filter frequency response. We use the generated coefficients to make the FIR filter in Verilog.

Figure 4-10 shows MATLAB simulation of GBM adaptation. We use a random speech signal as a target signal and a Gaussian noise signal. The signal and microphone setup are shown in Figure 4-9. The voice activity detector (VAD) signal from the mode controller allows GBM delay adaptation during speech duration. Initially, the beamformer is steered to 90° while the actual target signal comes from 110°. GBM adjusts the DOA, and TD_m converges well after 3s.

Furthermore, the GBM can also use a bandpass filter or any other operator instead of LPF to focus on specific signal characteristics. For example, GBM with a bandpass filter can selectively track a voice in a certain frequency range.

```
% MATLAB code

fs=16000;
d = fdesign.lowpass('N,Fc',10,fs/8,fs);
Hd = design(d);
```

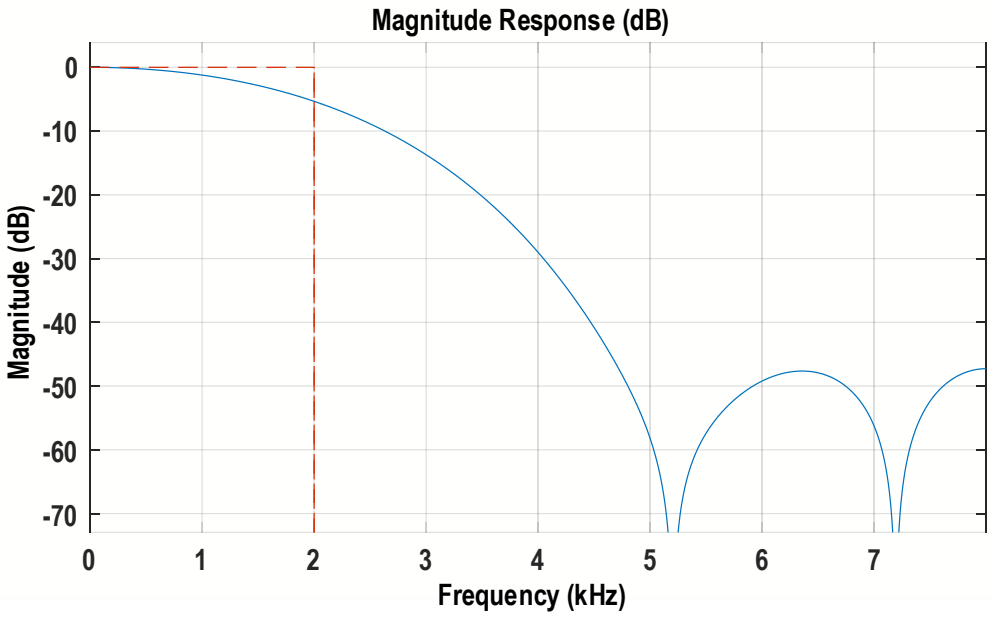


Figure 4-8. MATLAB code generates a low-pass filter for preventing the local minima issue and its frequency response.

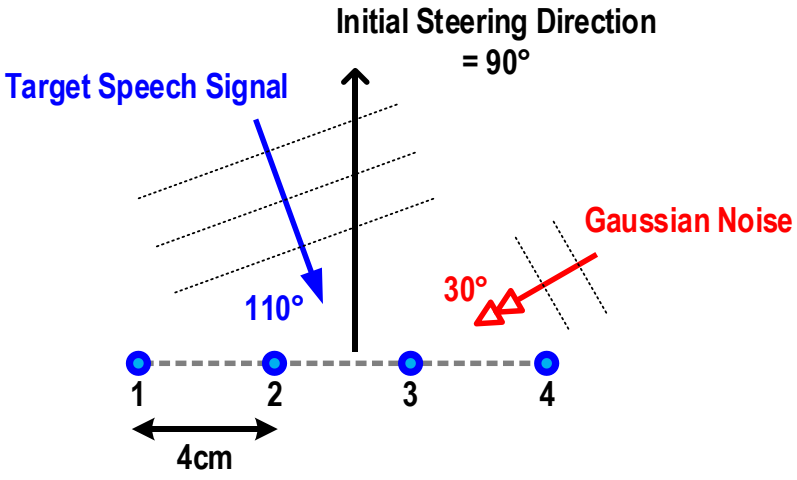


Figure 4-9. MATLAB simulation setup for Figure 4-10.

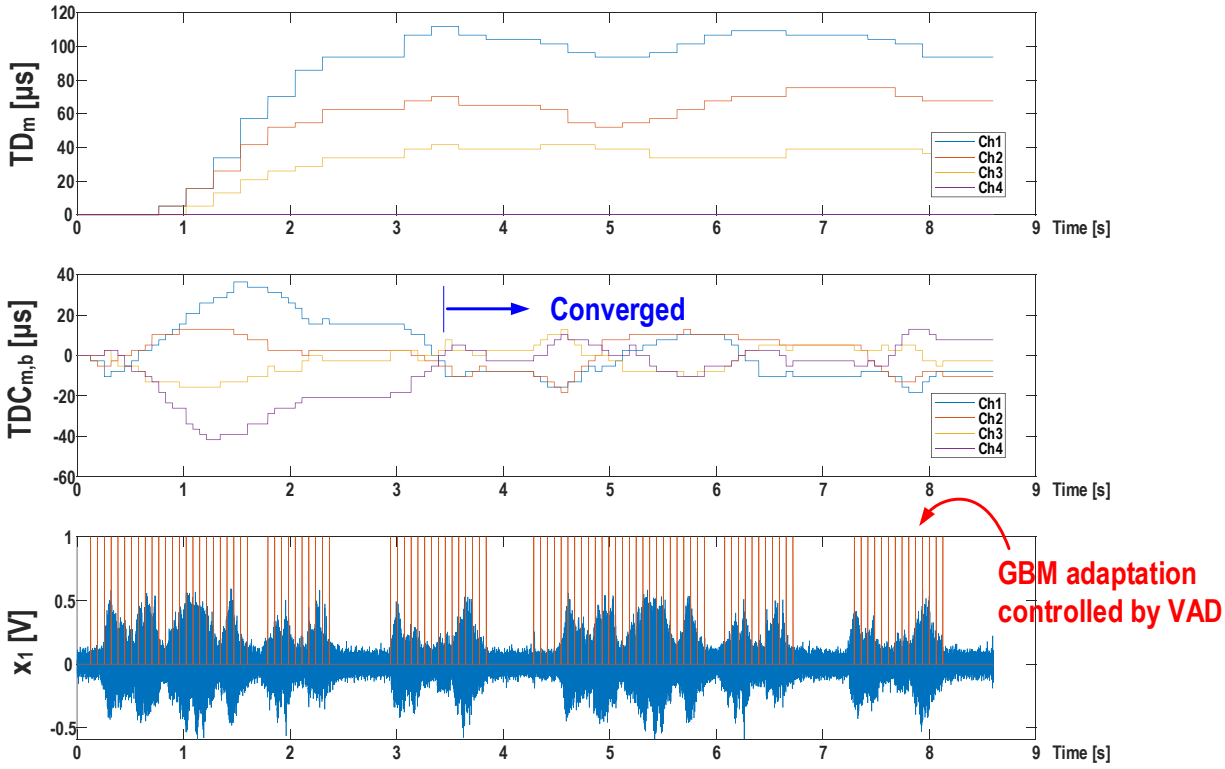
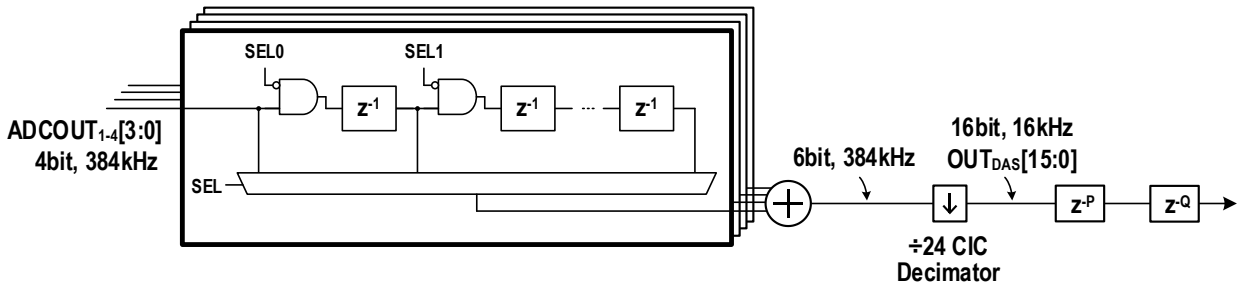


Figure 4-10. MATLAB simulation of GBM convergence. The input is a random speech signal.

4.2.2.2 DOA Tracking Delay and Sum Beamformer (DTDAS)

The proposed DTDAS (Figure 4-4) has two advantages over conventional DAS beamformers. First, it can correct initial TD_{1-4} by GBM feedback, as mentioned in the previous section. Second, it turns off unused delay cells to reduce switching power, as shown in Figure 4-11. (The actual schematic is different because it is implemented in Verilog in practice). Each channel has 120 delay cells composed of 4bit flip-flops operating at 384kHz.




```
always @(posedge clk or negedge rstb) begin
   if (!rstb) begin
      in_dly_n[0] <= 0;
      …
      in_dly_n[119] <= 0;
   end
   else begin
      in_dly_n[0] <= in;
      in_dly_n[1] <= in_dly_n[0];
      …
      in_dly_n[14] <= in_dly_n[13];

      if (sel==0) in_dly_n[15] <= 0;
      else in_dly_n[15] <= in_dly_in[14];
      …
      if (sel==104) in_dly_n[119] <= 0;
      else in_dly_n[119] <= in_dly_in[118];
   end
end
```

Turn off unused delay cells

Figure 4-11. A conceptual schematic of DTDAS and its pseudo-Verilog code.

4.2.2.3 Multiple-input Canceller (MC)

The proposed system improves the robustness of conventional MC when adaptation mode changes by using rollback coefficients. The generalized sidelobe canceller type beamformer (used in this work) adapts the BM and the MC in different signal situations. For example, the GBM adapts its coefficients when the target signal is strong, but the MC adapts when noise is strong. Hence, a system controller needs to sense the signal status (e.g., SNR estimator) and control the GBM and MC adaptation. The proposed system uses a VAD to determine adaptation; it adapts GBM when VAD=1 and adapts MC when VAD=0.

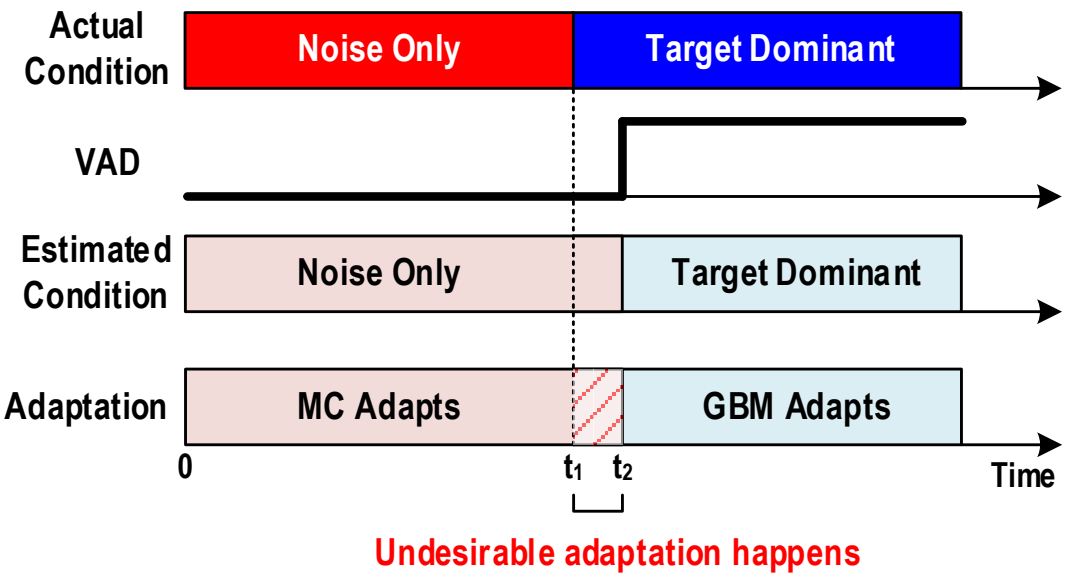


Figure 4-12. A mismatch between actual and estimated signal conditions and the adaptation of GBM and MC.

Adaptation timing is crucial for the overall beamformer performance. However, there is an inevitable timing error in VAD due to decision delay and estimation error. For example, assume an error in VAD exists between t_1 and t_2, as shown in Figure 4-12. The GBM has not started

adaptation from t_1 to t_2, even though the target signal is present. Then, y_{1-4} has target signal leakage from t_1 to t_2 due to the incomplete GBM adaptation. As a result, the MC experiences different signal conditions between $0 \sim t_1$ and $t_1 \sim t_2$ while adapting coefficients. Ideally, the MC should stop adaptation at t_1. If the VAD detector does not notice the change of signal condition, it may lead to an undesired change in the coefficients of MC at $t_1 \sim t_2$.

Figure 4-13 shows the simulated waveforms. Due to a wrong GBM adaptation from the wrong VAD, MC coefficients changes drastically during t_1 to t_2, as shown in the middle of Figure 4-13. To mitigate this unwanted behavior, our proposed MC rollbacks the coefficients at t_1 from t_2, as shown at the bottom of Figure 4-13. As a result, the MC coefficients keep their correct value after the mode change.

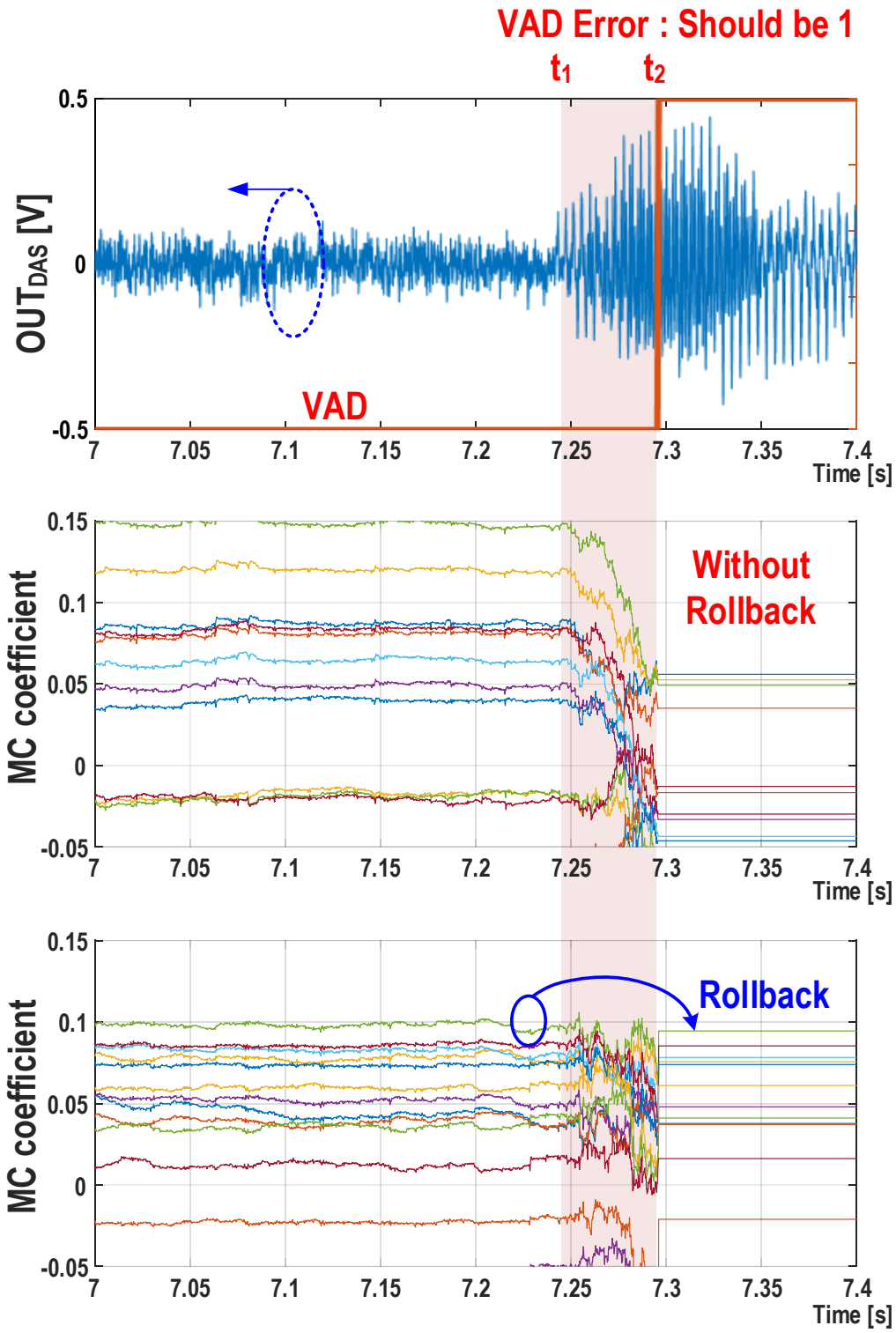


Figure 4-13. Simulated waveforms of the adaptation of MC from t_1 to t_2 in Figure 4-12 with/without rollback.

4.2.3. Multi-mode ADC

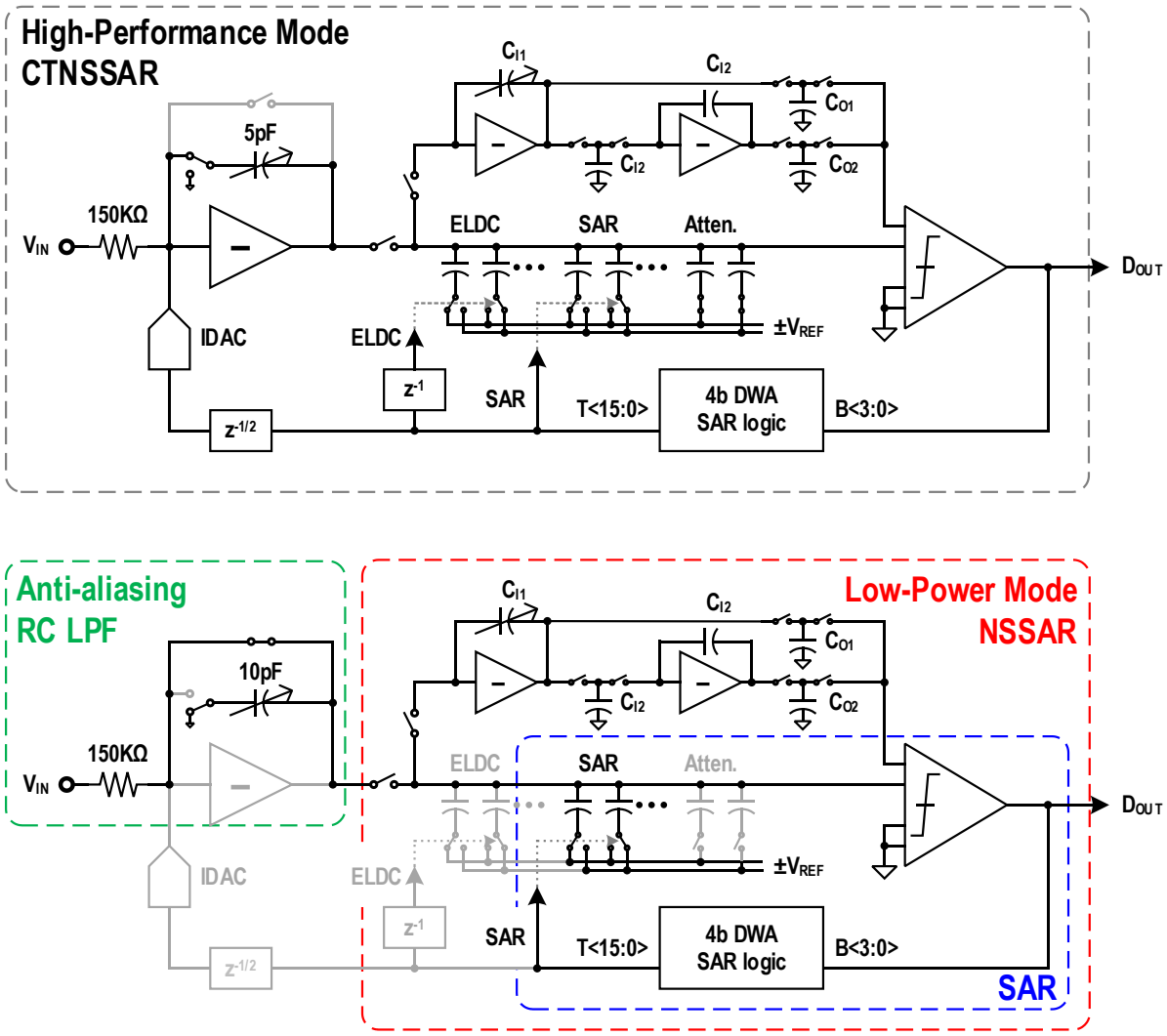


Figure 4-14. Multi-mode ADC showing (top) high-resolution operation with CTNSSAR hybrid and (bottom) low-power NSSAR mode and ultra-low-power SAR mode.

Multi-mode ADC operation and a hybrid architecture reduce ADC power by more than an order of magnitude. The system takes advantage of high signal SNR to adjust between 3 ADC modes to save power. However, a challenge is that conventional high-resolution ADCs cannot easily scale performance. The hybrid Continuous-time (CT) sigma-delta modulator (SDM) with a noise-shaping SAR (NSSAR) quantizer is 4x more efficient than the SDM in [18]. Furthermore, a

low ADC sample rate reduces the area and power of the DTDAS. 3rd order noise shaping and a 4-bit quantizer facilitate a low ADC sample rate of 384 kHz with OSR=24. We selectively operate blocks depending on the required SNR. The high-performance mode (80dBA SNDR 12μW) activates all blocks and works as CTNSSAR. The low-power mode (65dBA SNDR 5.8μW) enables only the NSSAR, while an ultra-low-power mode (40dB SNDR 1.5μW) uses only the SAR block within the NSSAR.

The high-performance CTNSSAR mode combines a 1st-order CT modulator loop and a 2nd-order NSSAR (Figure 4-14). An advantage of the CT loop is the inherent anti-aliasing filtering, but the two-stage CT amplifier (Figure 4-15) consumes half the total ADC power. The low-power mode turns off the CT loop and operates only the DT NSSAR block. The low-power mode re-uses the input resistor and capacitor of the CT stage as an RC low-pass filter to maintain anti-aliasing. Attenuation capacitors in the SAR ensure ADC-gain matching between different modes. Adjusting the NSSAR integrator capacitor, C_{I1} maintains noise transfer function (NTF). Dynamically-enabled folded-cascode amplifiers in the NSSAR save power [53]. A current feedback DAC (IDAC) is smaller and more efficient than a resistor DAC (Figure 4-16). Finally, the IDAC and CDAC share a DWA block to improve linearity.

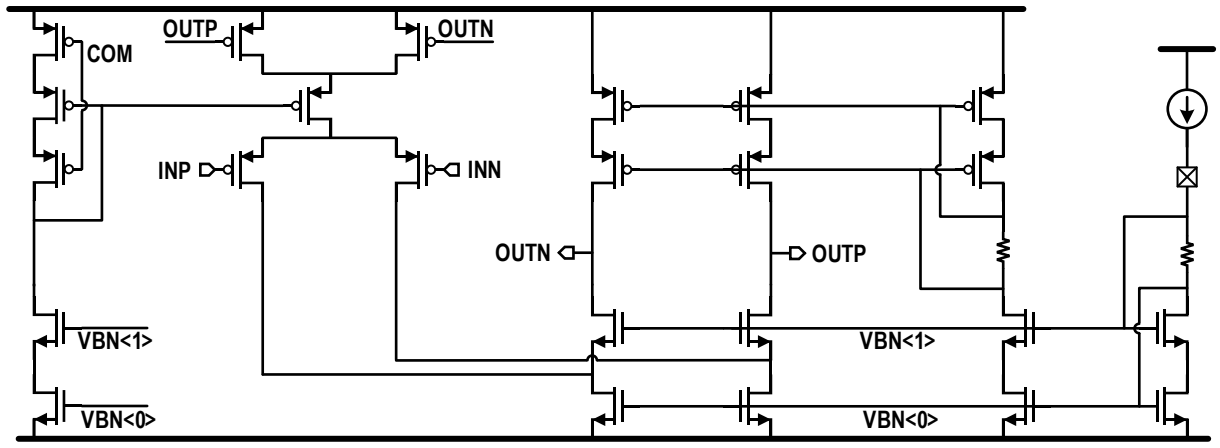


Figure 4-15. Schematic of 1st stage amplifier.

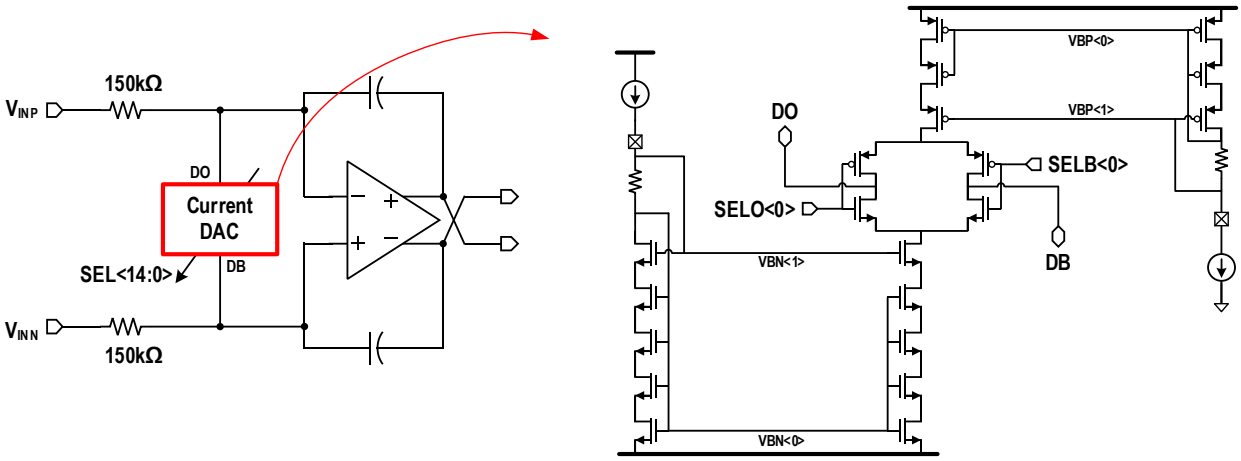


Figure 4-16. Schematic of current DAC (IDAC).

4.2.4. Mode Controller

The mode controller (Figure 4-1) of the proposed system controls three different modes: 1) GBM and MC adaptation timing controlled by VAD, 2) beamforming modes controlled by noise floor, and 3) ADC modes controlled by signal floor. Figure 4-17 shows a flow chart for the mode controller. In principle, our system can work with any VAD algorithm. The mode controller is fully implemented in Verilog and synthesized.

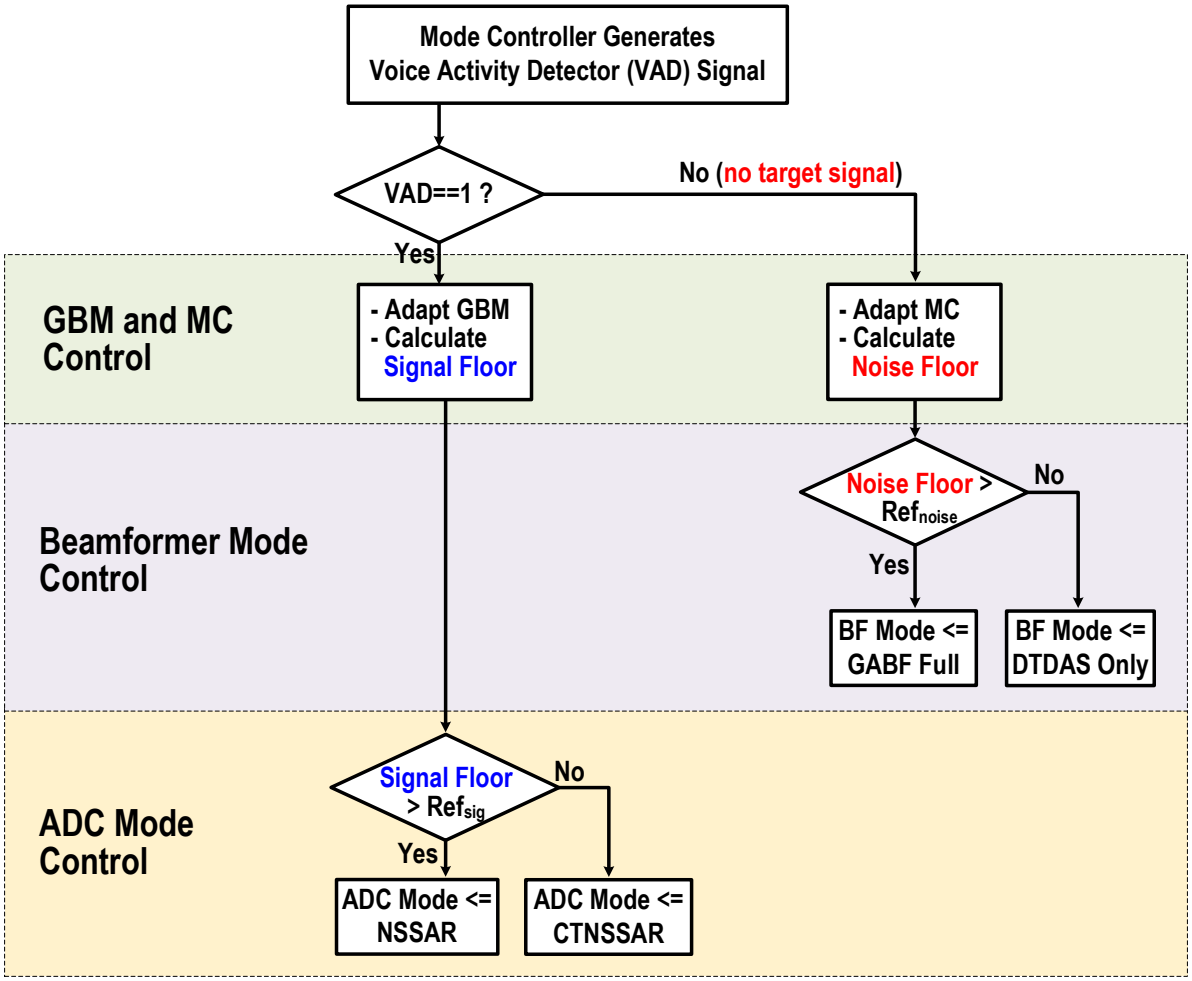


Figure 4-17. Flow chart of mode control.

4.2.4.1 VAD Generation

The mode controller uses an energy-based VAD for simplicity [51] - Figure 4-18 shows the calculation flow chart. Since speech energy primarily lies in low frequencies, our mode controller selectively chooses the output of the feature extractor (*Feature* in Figure 4-1) and calculates the signal floor. Figure 4-19 shows a simulated waveform, and VAD well indicates the existence of speech l. In the simulation, A=0.97, B=0.6, T_{DWN}=0.9, and T_{UP}=1.1 are used.

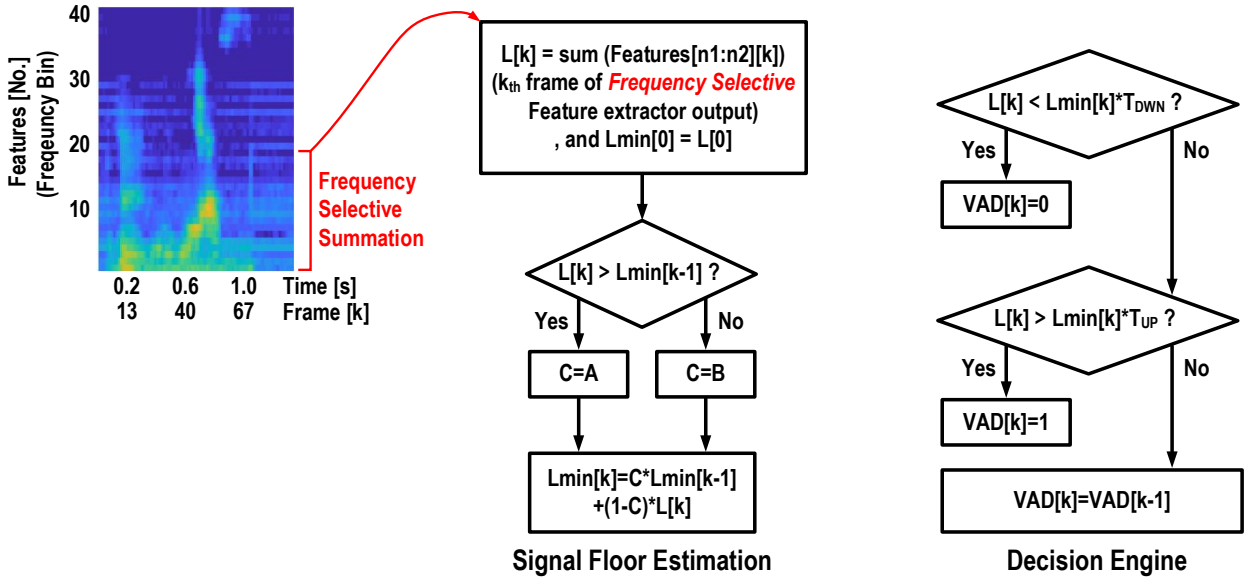


Figure 4-18. Flow chart of VAD signal generation [51] with proposed frequency-selective calculation.

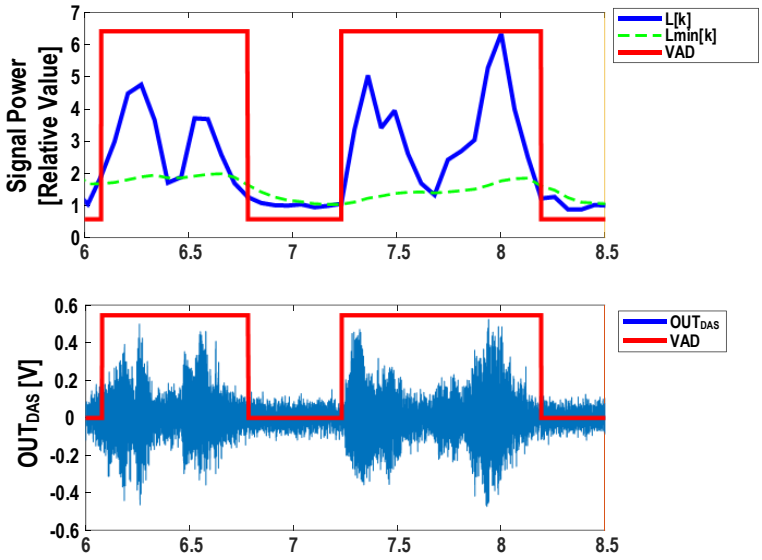


Figure 4-19. Simulated waveforms of VAD generation.

4.2.4.2 GBM and MC Control

The generated VAD solely controls the adaptation of the GBM and the MC. The primary purpose of the GBM is to filter out target speech signals from $x_{1\text{-}4}$ and generate noise-dominant $y_{1\text{-}4}$. Hence, the coefficients of the GBM ($TD_{1\text{-}4}$ and $TDC_{1\text{-}4,a\text{-}c}$ in Figure 4-4) must be adapted while there is a target speech signal. While speech exists (VAD=1), the mode controller activates GBM adaptation to remove the target signal from $x_{1\text{-}4}$ and generates $y_{1\text{-}4}$. Otherwise, while only noise exists (VAD=0), MC adapts its coefficients. Note that it assumes the noise source is relatively static while the target signal exists. For example, if a noise source spatially moves quickly during the VAD=1 situation, the previously adapted MC coefficients (based on noise source in a different position) do not effectively suppress the noise.

Figure 4-20 shows simulated waveforms of GBM and MC adaptation, where the input setup is shown in Figure 4-9. With the mode controller, OUT_{GABF} shows that GABF suppresses the noise well in x_1, and its actual sound confirms the effectiveness. TD and TDC adapt when VAD is 1, and MC coefficients adapt when VAD is 0.

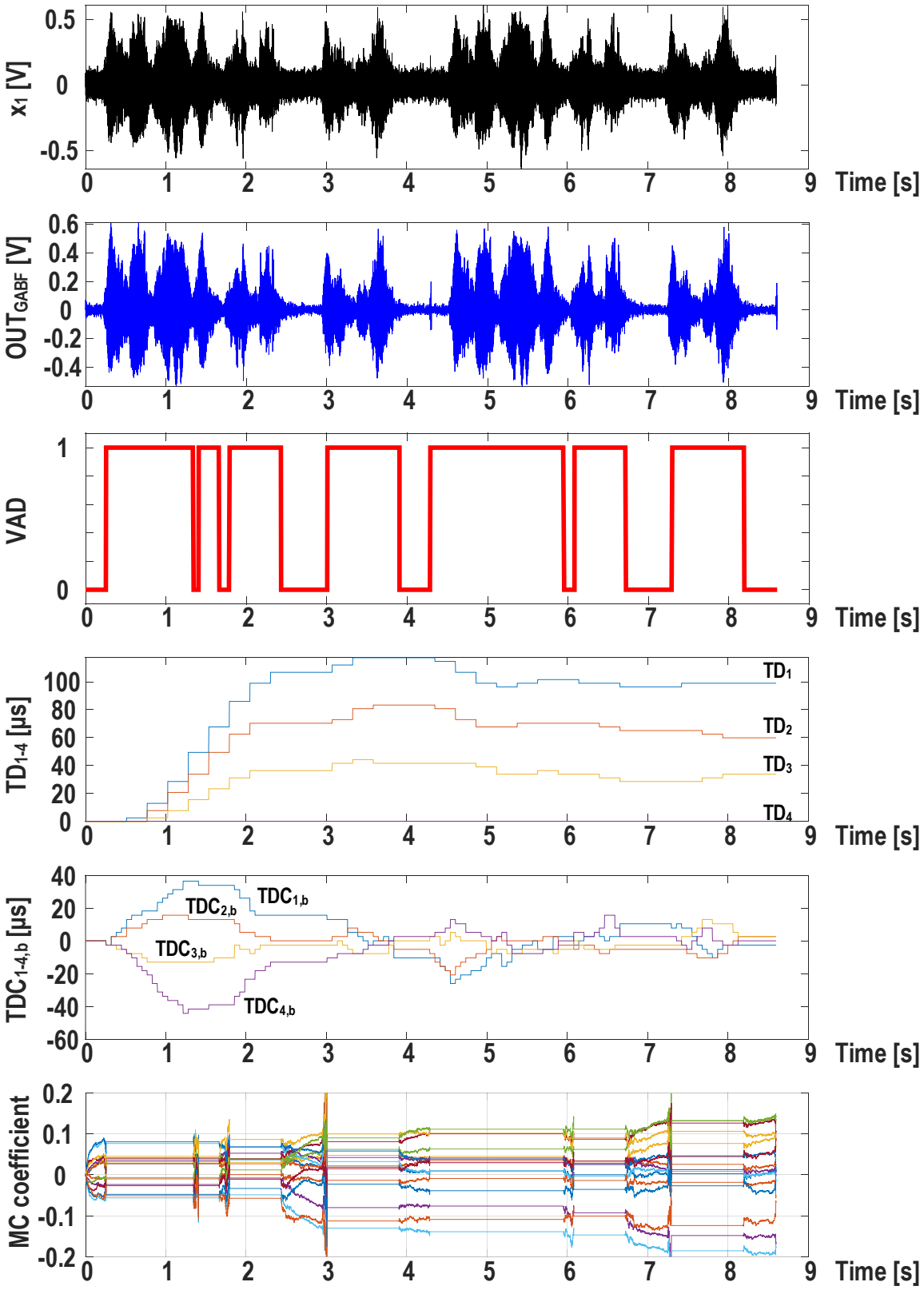


Figure 4-20. Simulated waveforms of GBM and MC adaptation controlled by VAD signal.

4.2.4.3 Beamformer Mode Control

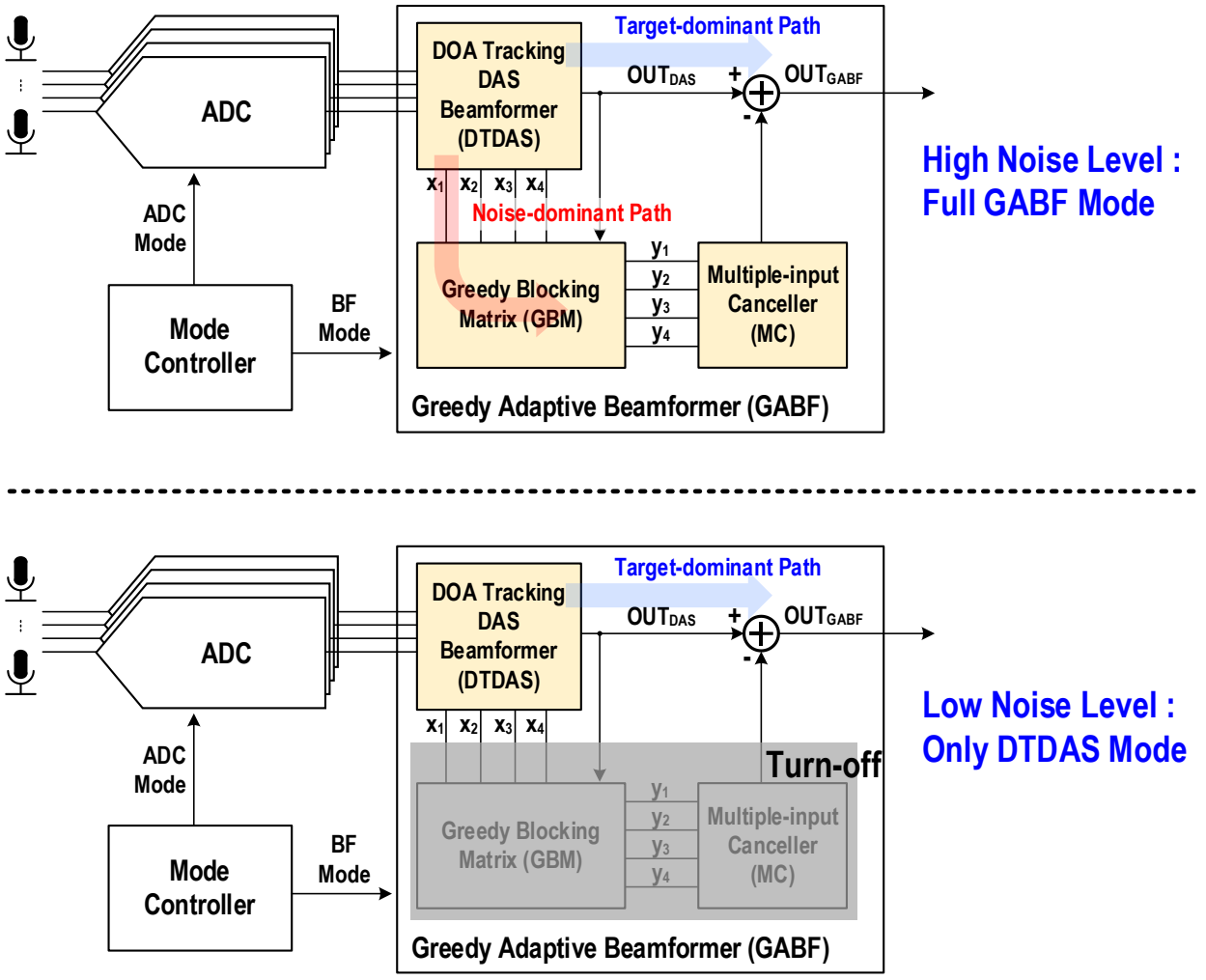


Figure 4-21. System diagram of the two-mode beamformer.

The proposed system has two beamformer modes: 1) full GABF mode using GBM and MC when the noise level is high, and 2) only DTDAS mode turning off GBM and MC when the noise level is low. The advantage of the RGSC structure is that it has two separate signal paths (target-dominant and noise-dominant paths); hence it is easy to turn off one of them without much effort [54]. Figure 4-21 shows system diagrams of two modes. For low-power mode, the mode controller turns off GBM and MC; hence it uses only DTDAS. In other words, the beamformer

works as DAS. In a low noise situation, the DTDAS alone can do enough job for noise suppression. To turn off GBM and MC, we short the inputs of GBM ($x_{1\text{-}4}$) to zero. Also, the calculation blocks use a 16kHz clock from the decimator to trigger operation. When we turn off the adaptive beamformer, we disable this clock disabling these calculations.

To estimate the noise level, we calculate the signal floor from OUT_{DAS} based on the algorithm in Figure 4-22. Whereas VAD uses only a low-frequency component for calculation, the noise floor uses the entire frequency span of signal OUT_{DAS} since the noise includes high-frequency. The mode controller calculates the noise floor of OUT_{DAS} only when VAD=0 (when only noise exists), and Figure 4-22 shows the flow chart of this calculation. Figure 4-23 shows the MATLAB simulated waveforms. The integration duration of OUT_{DAS} is 64ms in the simulation. The calculated noise floor increases when the background noise is stronger.

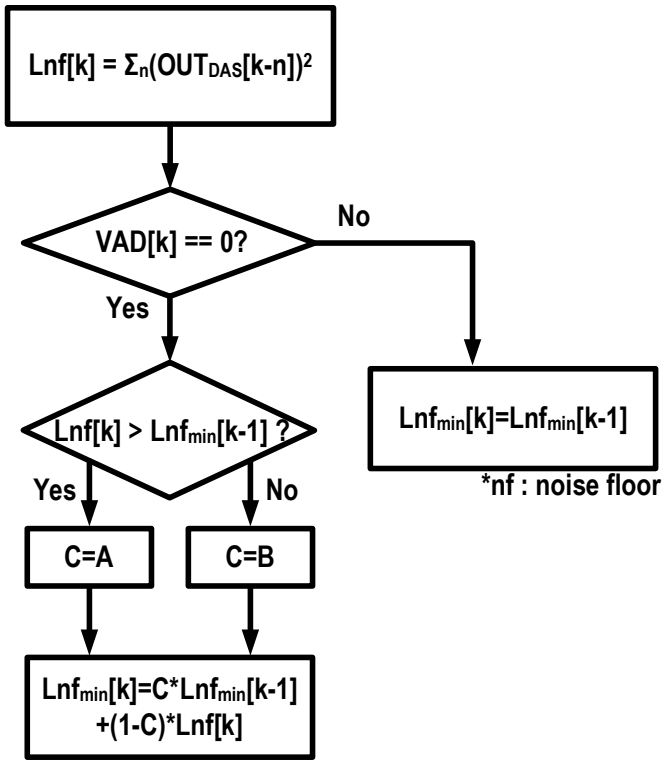


Figure 4-22. Flow chart of noise floor estimation based on [51].

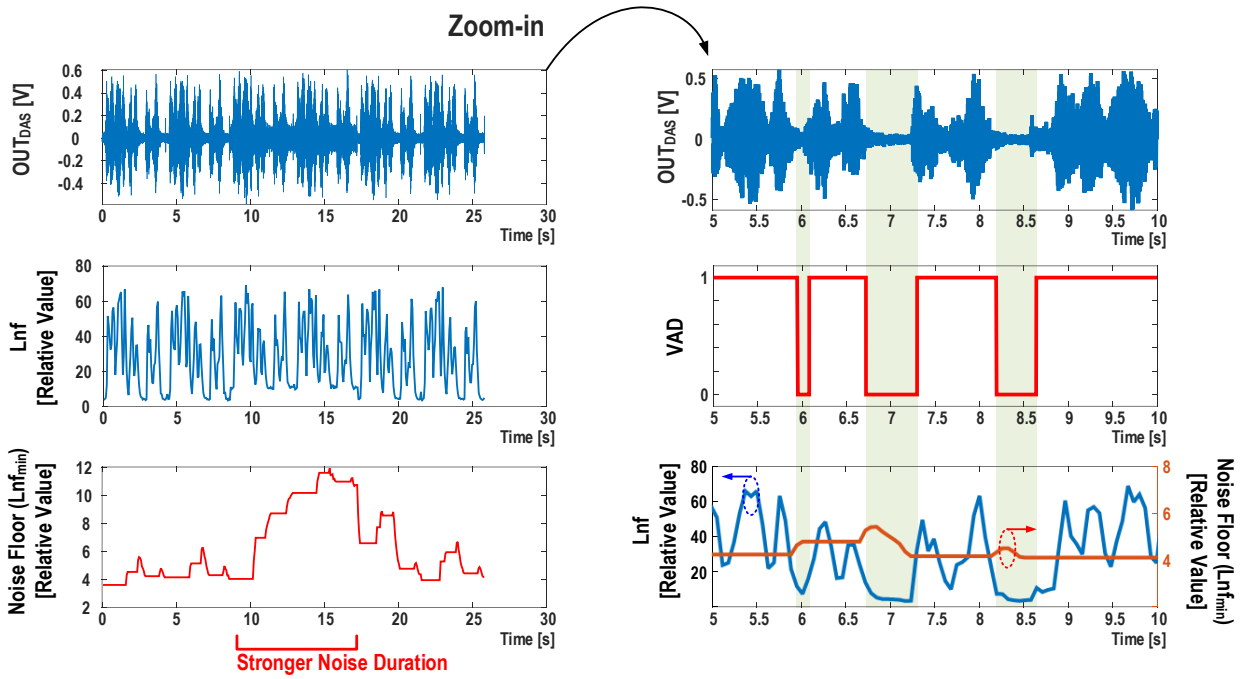


Figure 4-23. Simulated waveforms of noise floor calculation.

4.2.4.4 ADC Mode Control

As explained in the previous section, the proposed system provides three ADC modes: 1) High-performance mode CTNSSAR, 2) low-power mode NSSAR, and 3) ultra-low-power mode SAR. Our system changes ADC mode depending on the target signal level. For example, if the target signal is loud, the low-power mode NSSAR provides sufficient speech recognition accuracy. Figure 4-24 shows simulated speech recognition accuracy versus ADC effective number of bits (ENOB). First, we train a MATLAB DNN with a training set with 0dB signal power on average. Then, we sweep the signal power of a test set and the ENOB, and then check accuracy. When the input power is 0dB, ADC ENOBs of 7, 10, and 14 provide the same recognition accuracy. This indicates that when the input signal is large, a low-resolution ADC is sufficient. On the other hand, as the input power decreases, a larger ENOB provides better accuracy, validating the necessity of a high-resolution ADC when the input signal is small.

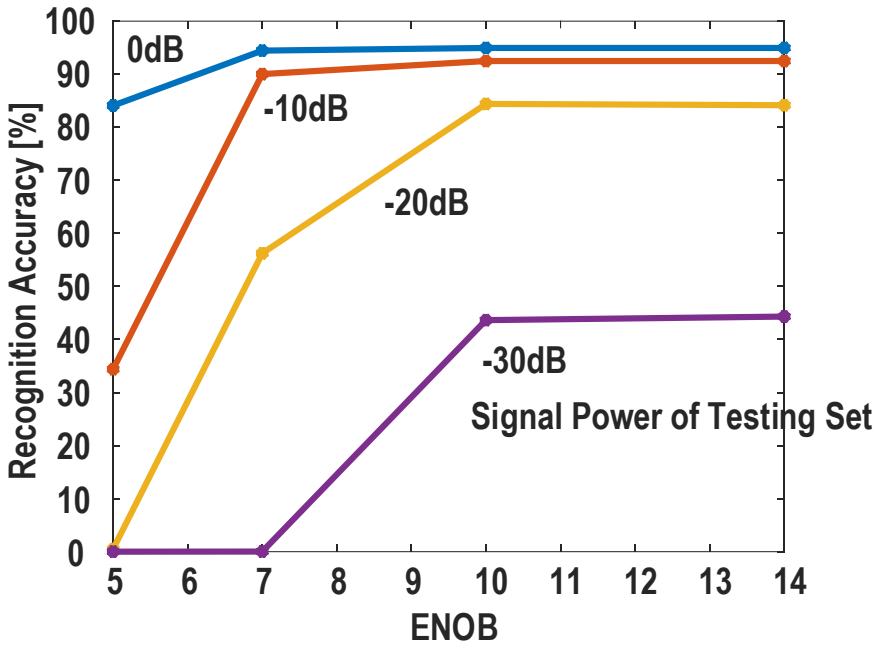


Figure 4-24. Simulated speech recognition accuracy Vs. ADC ENOB while sweeping input signal power.

Figure 4-25 shows a flow chart of signal floor calculation. Note that the mode controller calculates the signal floor when VAD=1 (speech exists). It uses the feature extractor output to select frequency bands - we select a low-frequency part as a default since it has the most speech energy. Note that the signal floor calculation performs twice and is averaged for accuracy. Figure 4-26 shows MATLAB simulation results, and Signal Floor (Lsf_{min}) represents the signal magnitude well.

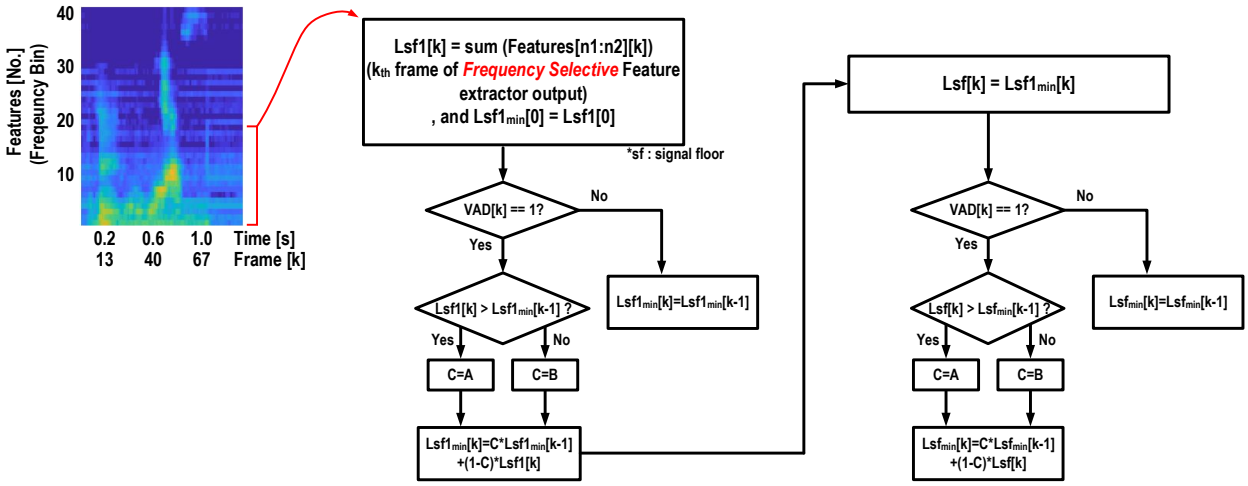


Figure 4-25. Flow chart of signal floor estimation based on [51].

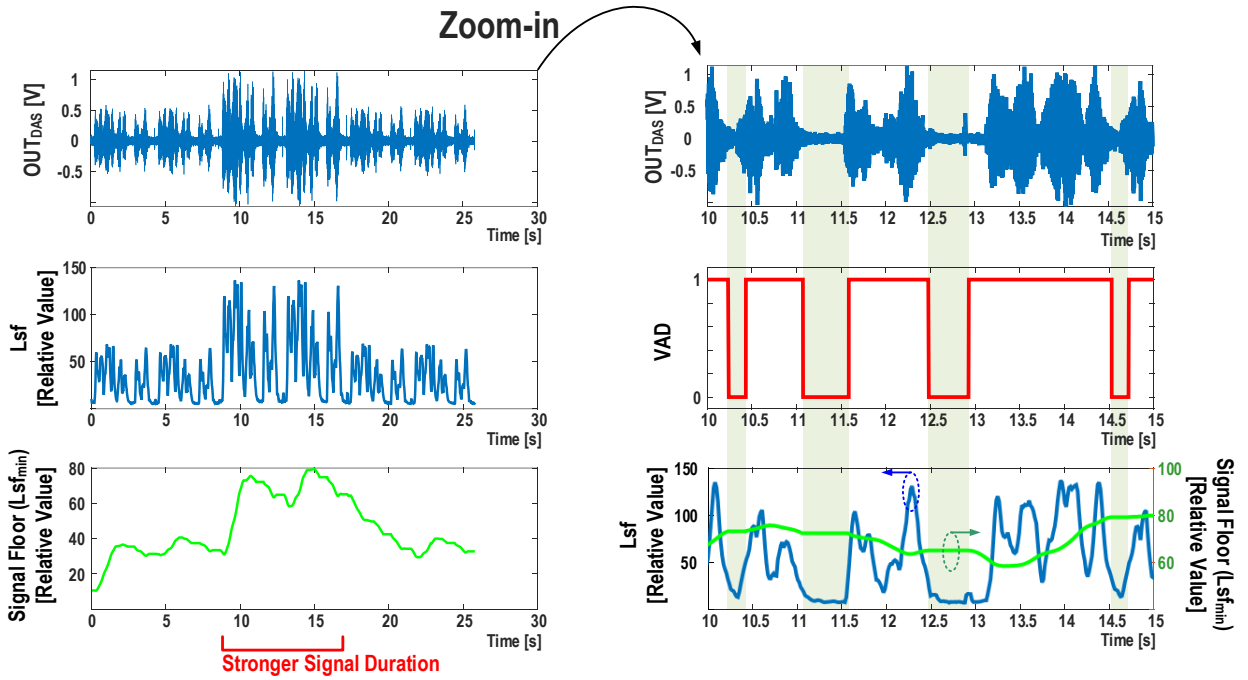


Figure 4-26. Simulated waveforms of signal floor calculation.

4.2.5. Feature Extractor

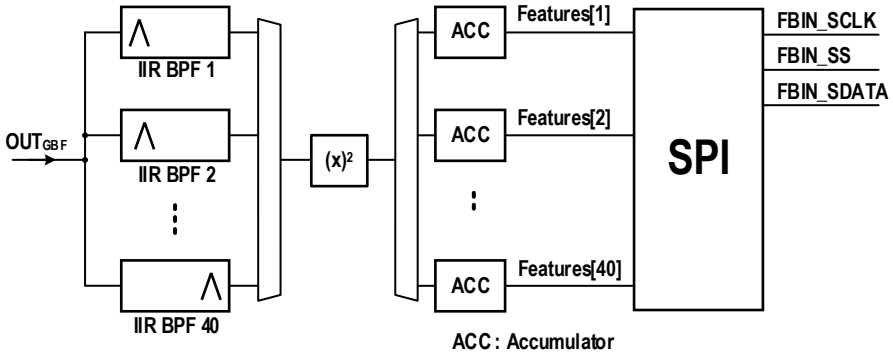


Figure 4-27. System diagram of feature extractor.

Figure 4-27 shows a system diagram of the feature extractor. We share a single squaring operator for the 40 bandpass filters to save area. To generate IIR bandpass filters, we use MATLAB designfilt function, where DesignMethod is butter and FilterOrder is 4. Then we convert the digital filter to a state-space representation and implement Verilog filters using manual code. Figure 4-28 shows the frequency response of the generated filters. The feature has a 25ms accumulation window with a 10ms overlap.

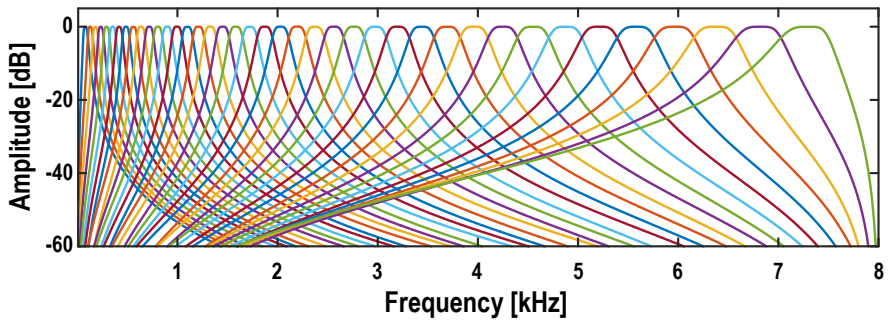


Figure 4-28. Frequency response of Mel-frequency filter-bank.

4.3. Measurements

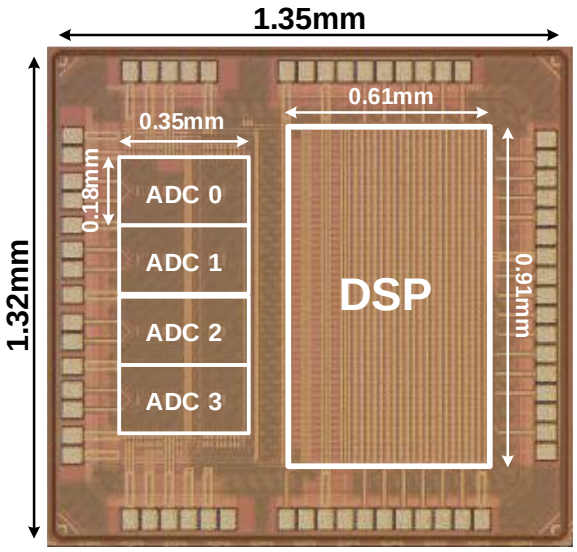


Figure 4-29. Die micrograph.

The prototype system is fabricated in 40nm LP CMOS (Figure 4-29) and occupies 0.94 mm^2. The total measured power in GABF mode with CTNSSAR ADCs is 157µW from 1V analog and 0.7V digital supplies. On the other hand, the low power mode with DAS and NSSAR ADCs consumes only 72µW.

4.3.1. Test Setup

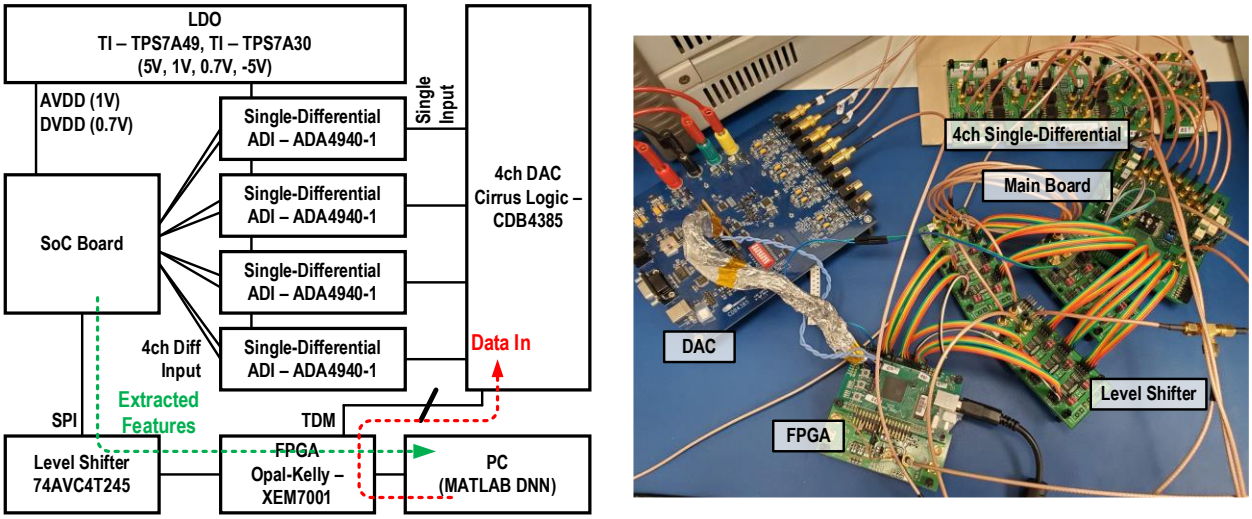


Figure 4-30. Board diagram and photo of the testing setup.

We use a 4-channel audio DAC (Cirrus Logic CDB4385) to emulate inputs from a cardioid microphone array. External single-to-differential amplifiers (Analog Devices ADA4940) convert the single-ended output of DAC to differential output with a 0.5V bias. An Opal-Kelly XEM7001 FPGA controls the DAC board and reads outputs from IC. The internal signals and feature output are monitored by a Serial Peripheral Interface (SPI). Manually coded python code controls FPGA.

4.3.2. Coefficient Adaptation Timing

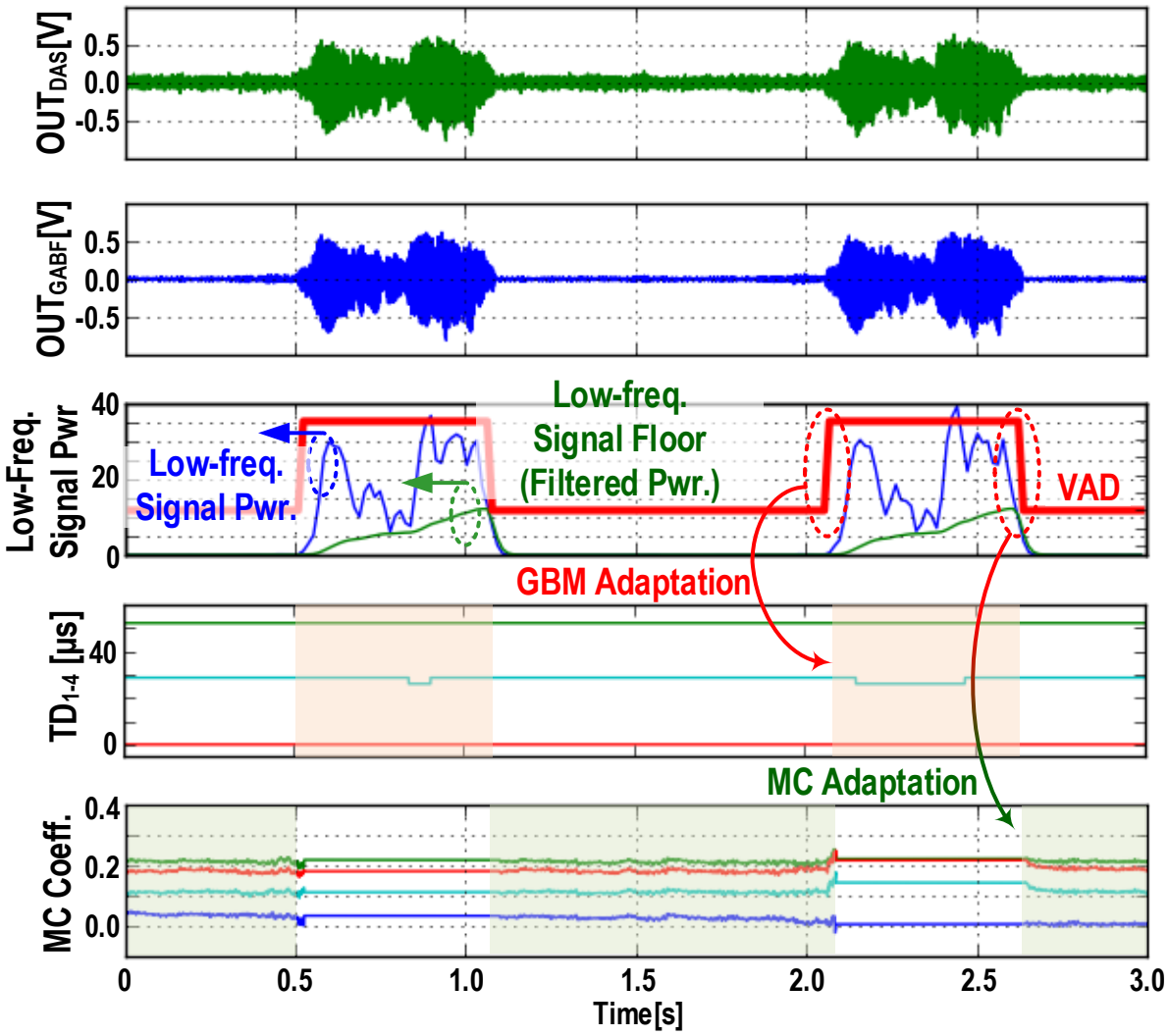


Figure 4-31. The measured waveform shows GBM and MC coefficient adaptation timing.

Figure 4-31 shows how the mode controller adapts GBM and MC depending on VAD. The mode controller calculates signal floor using the low-frequency features (since low frequencies dominate speech energy) and generates a voice activity detection (VAD) signal. The GBM adapts $TD_{1\text{-}4}$ when VAD is high (speech exists). The MC adapts the coefficients of the 28-tap FIR filters in each channel when VAD is low (i.e., dominated by noise). The bottom waveform shows partial four MC coefficients out of a total of 112.

4.3.3. GBM Adaptation (DOA Tracking)

We test the GBM adaptation with a 2kHz single tone sinewave with a linear microphone array to illustrate DOA tracking, as shown in Figure 4-32. We choose a linear configuration because it shows more intuitive results than a cardioid configuration. Initially, the given DOA is 90°, and TD_{1-4} are set to zero. However, the actual signal is coming from 70°. Since the input sinewave is assumed as a far-field signal, we can calculate the desirable TD_{1-4} using the equations from Figure 4-33. For example, microphone 1 receives a signal later than microphone 4 by Δt_1. Hence, assigning Δt_1 to microphone 4 will align the target signal between 1 and 4.

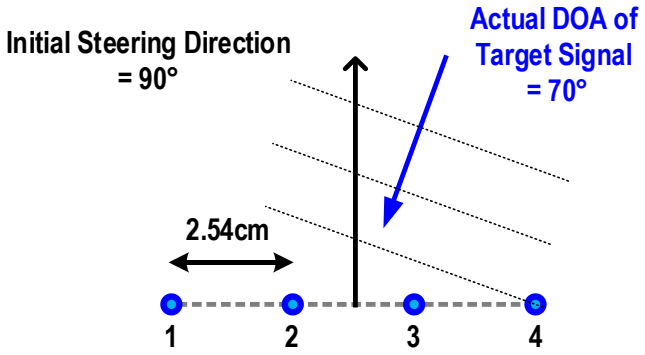


Figure 4-32. Microphone configuration for DOA tracking testing.

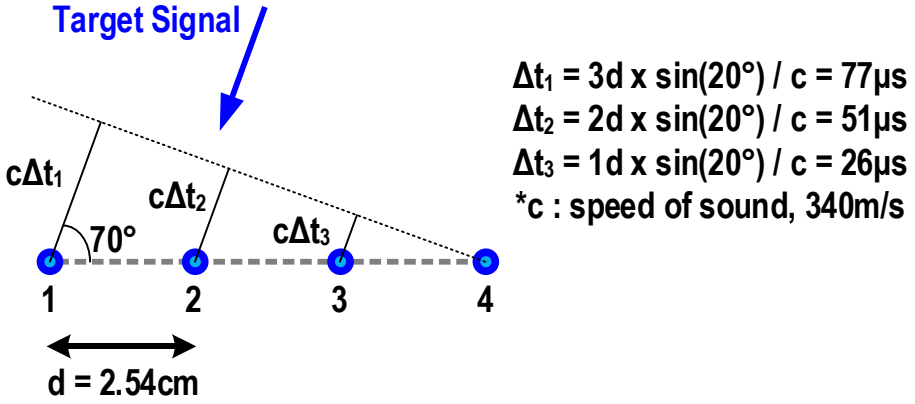


Figure 4-33. Delay of signal arrival for each microphone. For example, microphone 1 receives the target signal slower than microphone 4 by Δt_1.

Figure 4-34 shows the measured waveforms for GBM adaptation. In the beginning, TD1-4 are all zero since the initial steering angle is 90°. The top-right waveform shows a 2kHz sinewave received by the four channels. It shows a 77μs phase difference as calculated in Figure 4-33. Then, we enable GBM adaptation at t=0.2s. $TDC_{1\text{-}4,b}$ tries to align the signal by using the greedy algorithm. Then, GBM feeds back $TDC_{1\text{-}4,b}$ to $TD_{1\text{-}4}$ of DTDAS. After t=0.6s, the $TD_{1\text{-}4}$ arrive at the correct delay values, and $TDC_{1\text{-}4,b}$ settle to zero, indicating the DOA correction is complete. As a result, the input sinewaves are aligned, as shown in the bottom-right of Figure 4-34.

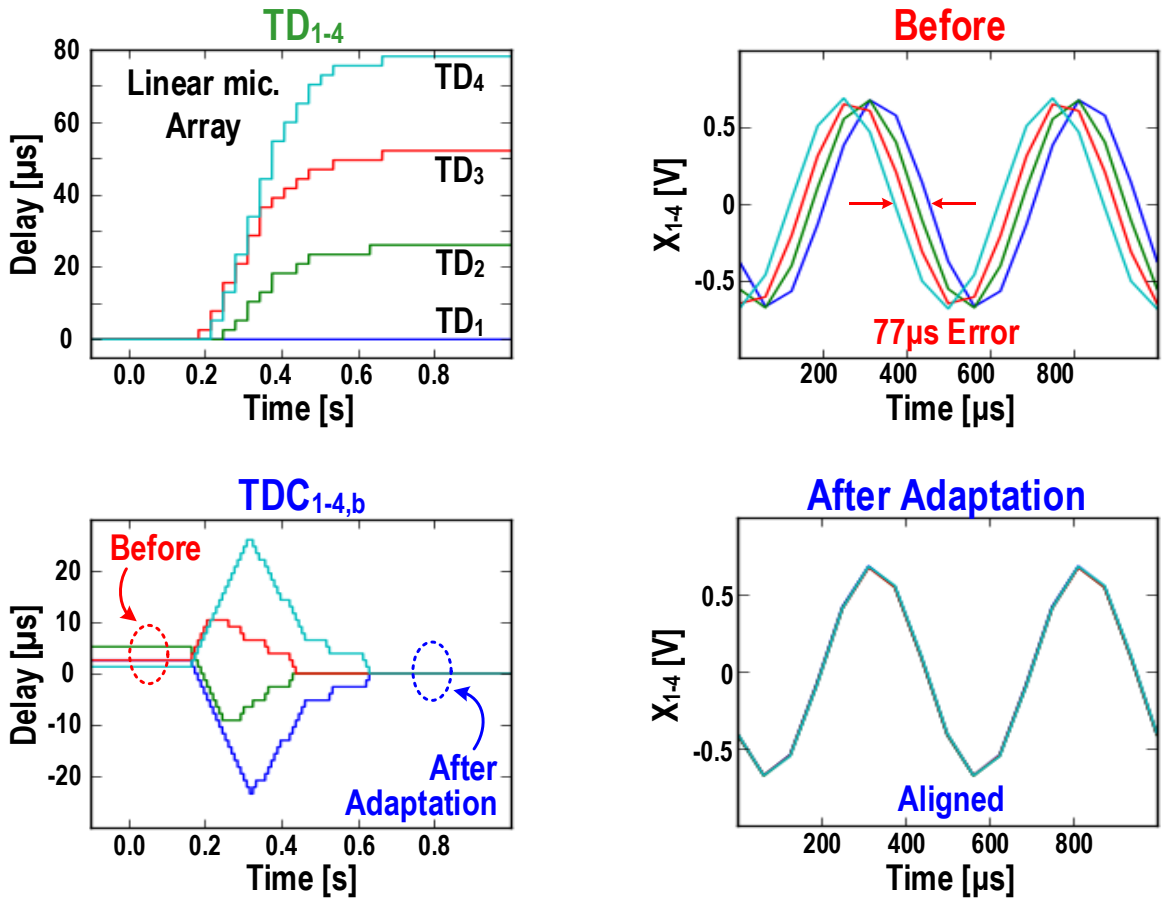


Figure 4-34. The measured waveforms of GBM adaptation. The proposed GBM adjusts 20° of DOA error by adapting $TD_{1\text{-}4}$ and $TDC_{1\text{-}4,b}$ as shown in the left two waveforms. The waveforms on the right show the alignment of signals after the adaptation.

4.3.4. ADC Measurements

The measured ADC SNDR is 79.6 dBA and 65.4 dBA in CTNSSAR and NSSAR modes. Figure 4-35 plots 32k point FFTs. The input is a 2kHz sinewave. Note that the cliff at 8kHz in FFT comes from the A-weight function.

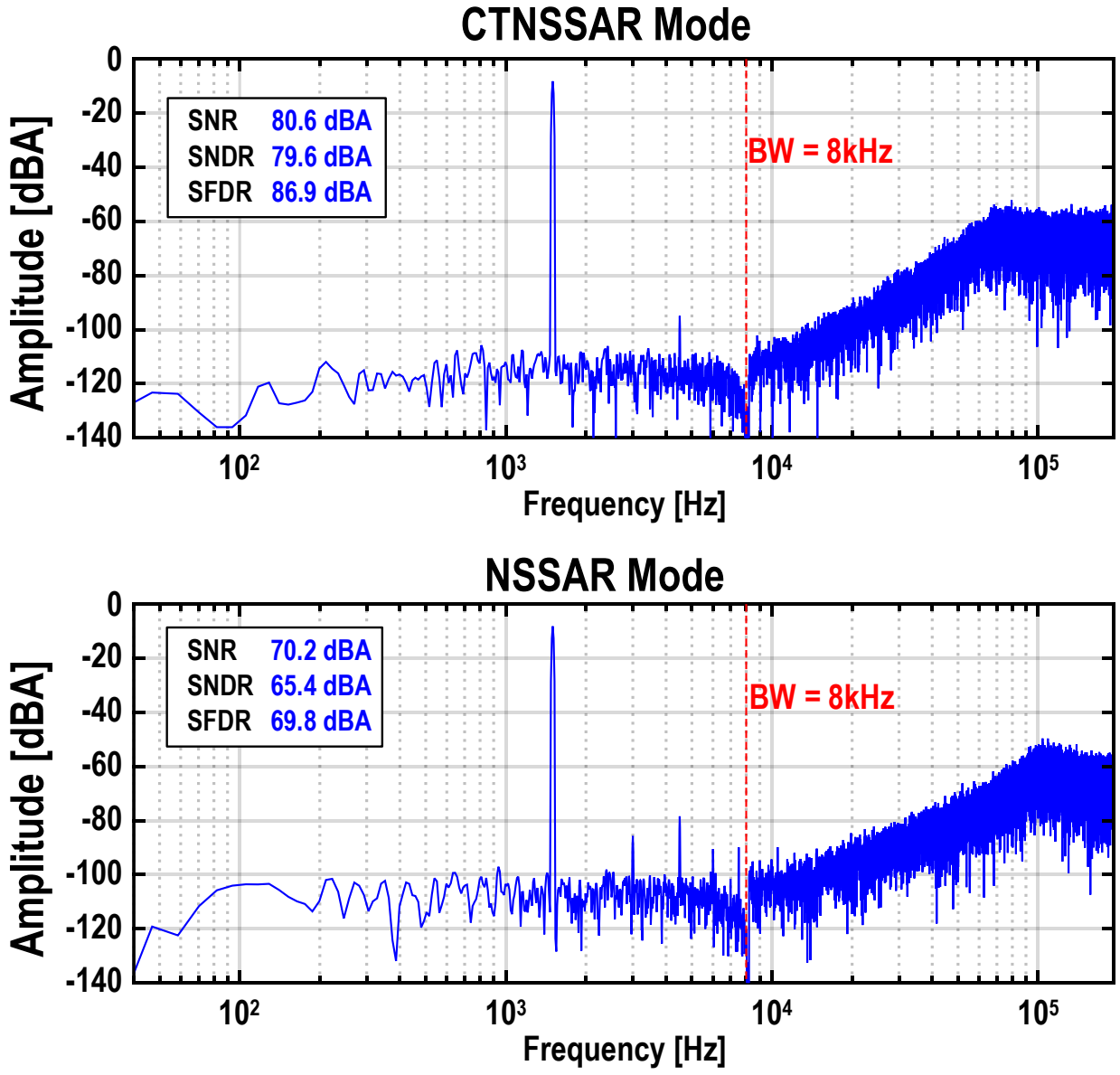


Figure 4-35. Measured 32k point FFT for single ADC in CTNSSAR and NSSAR modes

Analog ADC		
Supply [V]	1.0	
Area [mm^2]	0.18x0.45	
f_s	384kHz	
Bandwidth	8kHz	
OSR	24	
Input Impedance	150kΩ	
Mode	SNDR [dBA]	Power [μW]
CTNSSAR	79.6	12
NSSAR	65.4	5.8
SAR*	40	1.5

* Oversampling (OSR=24) 4bit SAR

Figure 4-36. Performance summary of ADC.

4.3.5. Mode Change

Figure 4-38 shows how the system adapts mode to the target power and noise floor. This test applies a random conversation (i.e., the target) at a 90° DOA and Tensorflow background noise at 30°, as shown in Figure 4-37. We repeat the same conversation source in Zone1-3 (Figure 4-38) while changing speech and noise power.

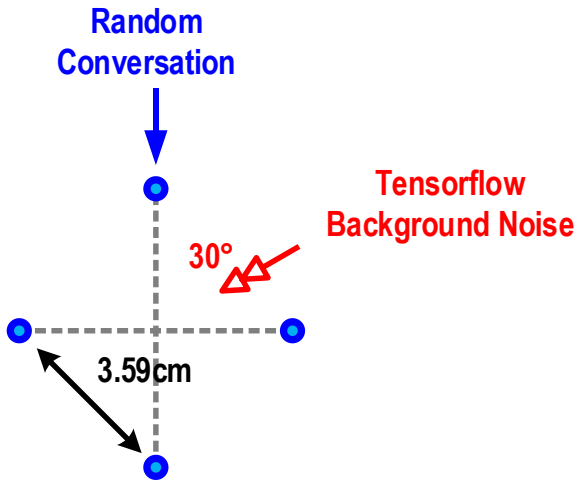


Figure 4-37. Microphone configuration and input signal direction for mode change measurement.

In zone 1 of Figure 4-38, the noise is stronger than in other zones. Hence, the estimated noise floor by the mode controller (shown in the bottom waveform of Figure 4-38) is high, and the system fully activates the beamformer. We see that the system effectively suppresses noise in the beamformer output, OUT_{GABF}. In this mode, the DSP consumes 109μW, and ADC consumes 48μW.

In zone 2, we lower the noise level while keeping the target speech power the same. As a result, the noise power decreases, so the mode controller turns the GBM and the MC off, and only DTDAS operates. As a result, DSP power reduces from 109μW to 49μW. One possible drawback of turning off the GBM and the MC is that the beamformer no longer has a DOA tracking feature. Thus, depending on the application, a user might fully operate GABF regardless of the noise conditions.

In zone 3, we increase the target speech amplitude while keeping the noise level the same. Hence, the estimated target power increases, allowing the ADCs to switch to low-power mode (NSSAR only). As a result, the analog power decreases to 23μW from 48μW, saving 25μW. The zoomed-in waveform in the inset shows the smooth mode transition in the ADC output.

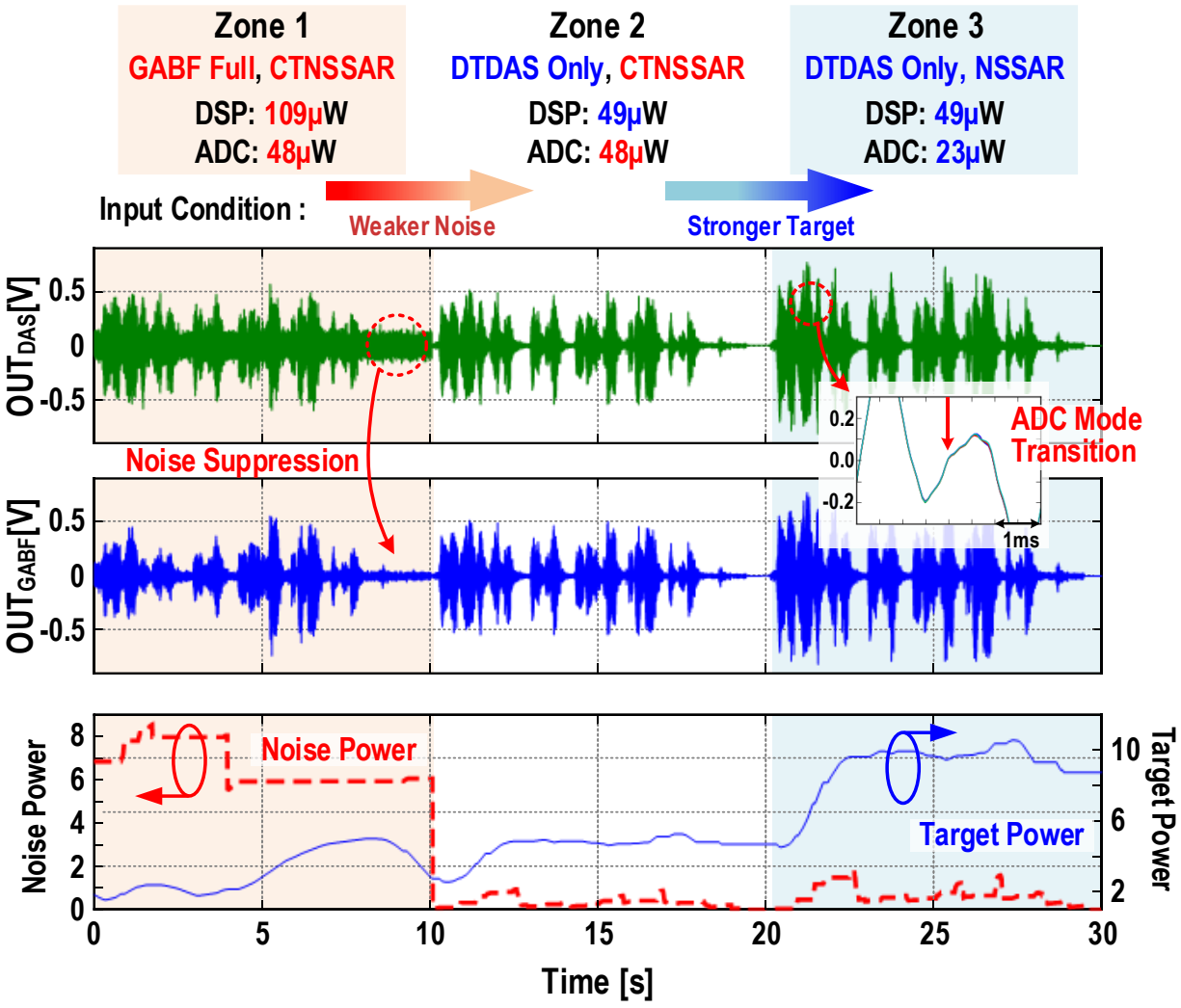


Figure 4-38. Measured adaptation to target power and noise floor.

4.3.6. Measured Beampatterns

Figure 4-39 shows measured beampatterns for different target and noise environments assuming a 3.59 cm-spaced cardioid microphone array. To measure beampatterns, we adapt coefficients of MC first with the target and two noise signals. Both target and noise signals are Gaussian signals. Meanwhile, we assume the steering angle is accurate with the actual DOA of the target signal; hence we assign the corresponding $TD_{1\text{-}4}$ to DTDAS and all 0s to $TDC_{1\text{-}4,b}$.

The target signal is a single-tone sinewave, and we sweep the DOA and measure the power of single-tone output using FFT. The proposed beamformer suppresses the noise well by placing nulls while maintaining the near-unity gain of the target direction, as shown in Figure 4-39.

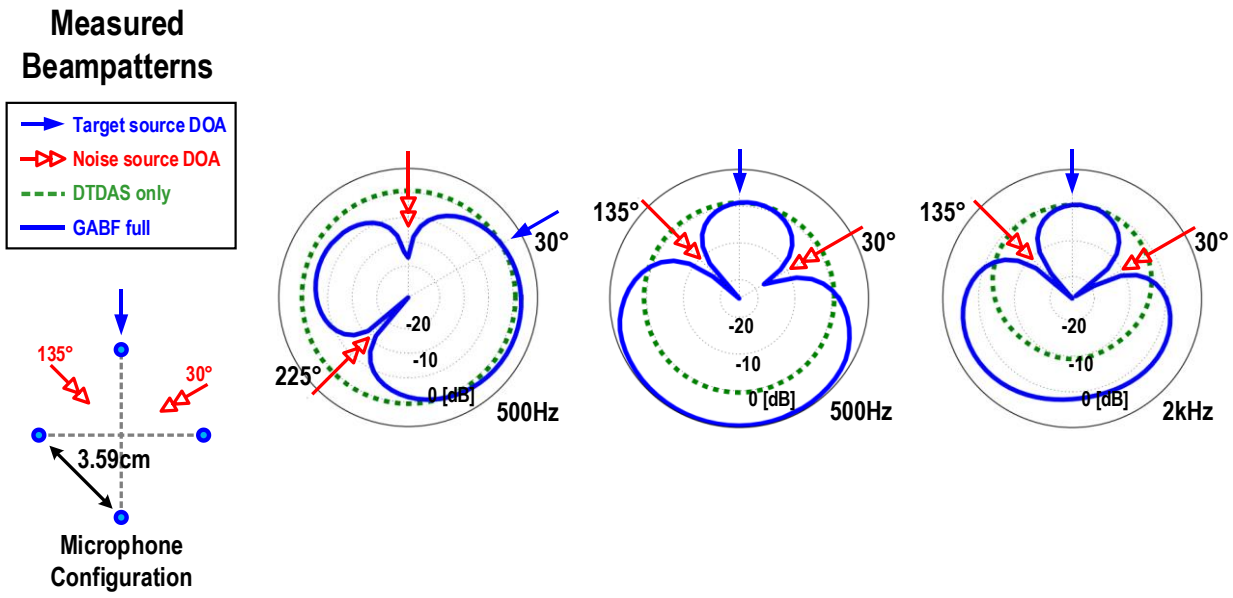


Figure 4-39. Cardioid microphone configuration (left bottom), measured beamforming patterns for proposed adaptive beamforming (GABF), and fixed DAS with different target and noise directions.

4.3.7. Speech Recognition Test

We use the Tensorflow speech dataset to validate the advantages of GABF for speech recognition. A MATLAB DNN with five convolutional layers and a single fully connected layer processes the chip spectrogram output [47]. The dataset includes 16 different words (with and without added random noise) and background noise. We train the DNN with feature extractor outputs from the prototype. The recognition accuracy without noise is 94.4%. Next, we test with interfering noise and interfering random background words to show the benefits of GABF. The interfering noise (or random words) are from DOAs of 30° and 135° and 9dB lower in power than the target with a cardioid microphone array.

The spectrograms in Figure 4-40 indicate that the GABF effectively suppresses noise. When the added noises are Tensorflow random background noise, full GABF beamforming improves the measurement recognition accuracy from 76% to 89%. The accuracy without beamforming is already high as 76% since the DNN is capable of handling the same kind of noise as used in training.

On the other hand, if the added noises are random words, GABF full beamforming significantly improves the accuracy from 54% to 83% (the overall accuracy is lower with word interference because the DNN is trained with added background noise). The accuracy without beamforming is 54% which is lower than that of testing with background noise since it is a different type of noise source used in training. However, the DTDAS beamforming alone only increases the accuracy by 3% due to its lack of noise suppression. Figure 4-41 shows its measured confusion matrix.

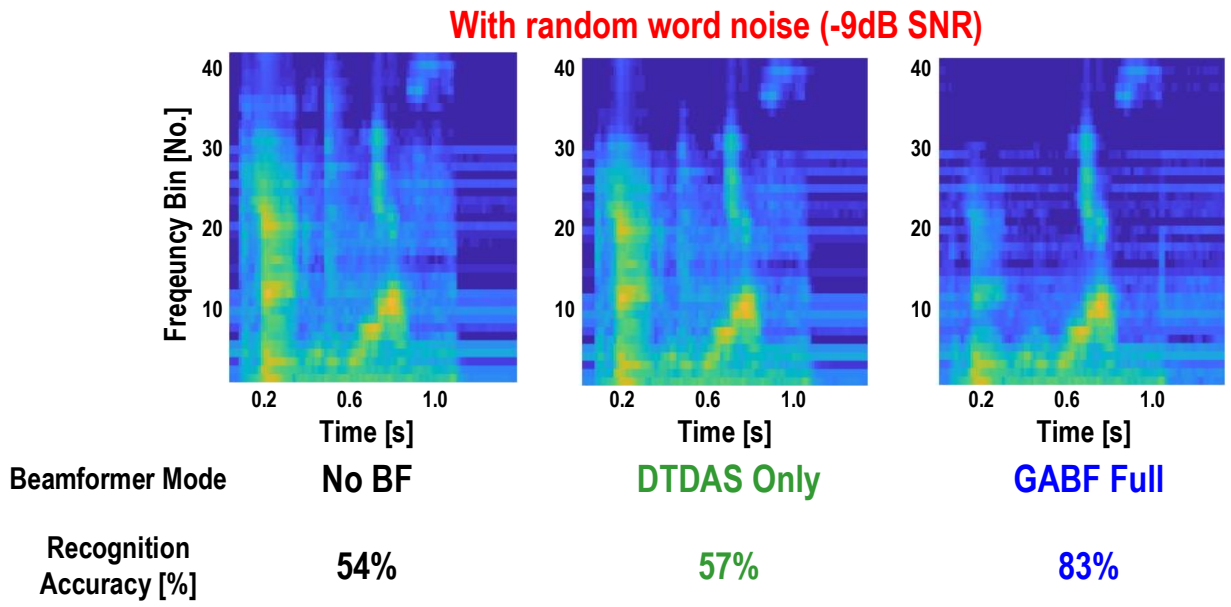


Figure 4-40. Measured spectrogram generated by the chip with different beamformer modes.

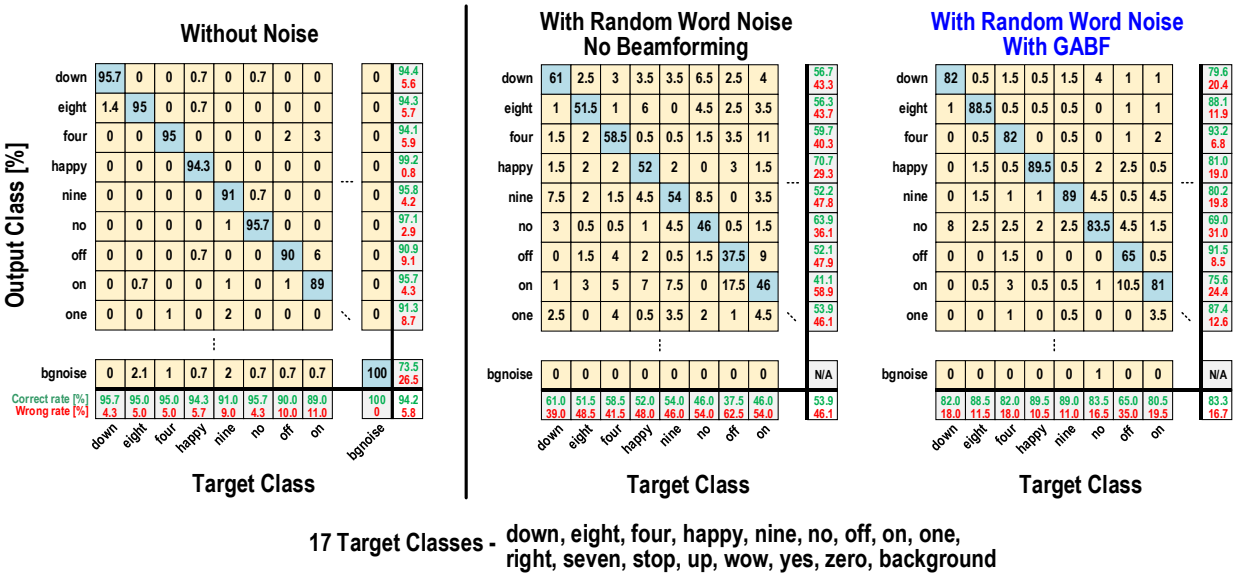


Figure 4-41. Measured speech recognition confusion matrix of: without noise, with random word noise with/without GABF beamforming. The prototype beamformer increases the recognition accuracy from 54% to 83%.

4.3.8. DSP Power Consumption Analysis

Figure 4-42 shows a power breakdown of DSP. There are two operation phases when GABF is fully turned on; MC adaptation or GBM adaptation (VAD signal controls their adaptation as discussed in mode controller session). When MC adapts, the whole DSP consumes 168 µW. MC consumes a majority of the power due to its FIR filter and coefficients calculation. On the other hand, when GBM adapts, the power consumption of MC significantly decreases since the coefficient calculation stops. As a result, the DSP consumes 50 µW. To represent the total GABF power consumption, we average the MC and the GBM adaptation power to get 109 µW. In DTDAS only mode, the GBM and the MC are turned off, and the DSP consumes 49 µW.

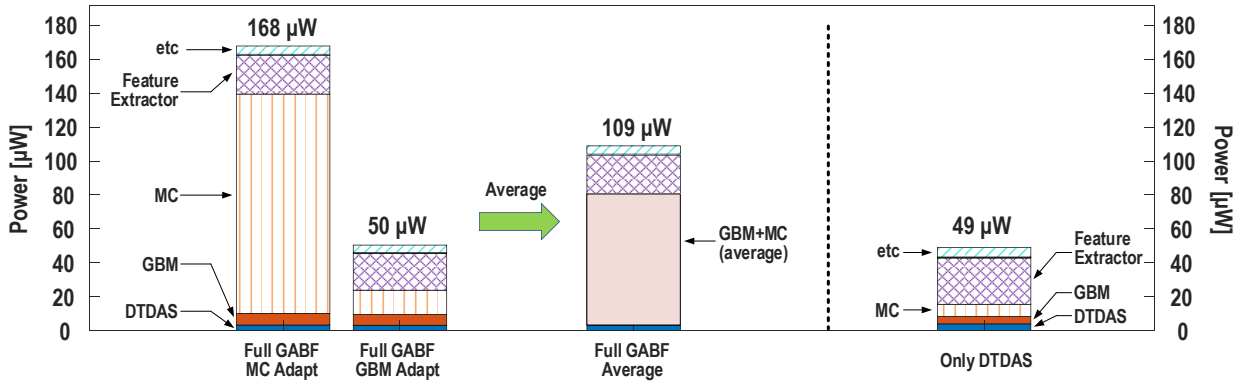


Figure 4-42. DSP power breakdown for different modes.

Table 2: Comparison with high-SNDR beamforming feature extraction systems.

	This Work	[4] Kang	[1] Lee	[8] Liu	[3] Sainath	Google Home
Implementation	**Analog mic. +Single chip**	Analog mic. +Single chip	Analog mic. +Single chip	Digital mic. +Multichip	Digital mic. +Software	Digital mic. +Multichip
Technology	**40nm LP CMOS**	40nm LP CMOS	40nm GP CMOS	90nm CMOS	-	-
Area (mm^2)	**0.94**	0.89	1.1	0.47	N/A	N/A
VDD (Analog / Digital)	**1.0V / 0.7V**	1.0V / 0.7V	1.0V / 0.55V	- / 0.33V	-	-
# Signal Sources	**4**	4	8	2	2	2
Functionality	**ADCs, adaptive beamforming, feature extraction**	ADCs, adaptive beamforming, feature extraction	ADCs, fixed beamforming, feature extraction	Adaptive beamforming (fixed steering), feature extraction	Adaptive beamforming, feature extraction, classification	ADCs, beamforming, feature extraction, classification
DR [8kHz BW]	**80 / 65dBA***	83dBA	85dBA	-	-	108dBA
BW	**8kHz**	8kHz	8kHz	8kHz	8kHz	8kHz
Beamforming Type	**Adaptive GABF**	Adaptive RGSC	Fixed delay-and-sum	Adaptive Griffiths-Jim	Adaptive filter-and-sum with trained coefficients	-
DOA Correction	**Yes**	No	No	No	N/A	N/A
Multi-mode Operation	**Yes**	No	No	No	No	N/A
Feature Type	**Log-Mel filter bank energy**	Log-Mel filter bank energy	Log-Mel filter bank energy	FFT-based Log filter bank	Convolutional long short-term memory DNN filter bank	-
# Features	**40**	40	60	8	128	-
AFE Power Consumption	**48 / 23µW***	367µW	0.81mW	-	-	-
DSP Power Consumption	**109 / 49µW****	280µW	3.1mW	0.1mW***	-	4.4mW****

*CTNSSAR / NSSAR, **GABF full / DTDAS only, *** Excludes ADCs, **** Calculated from datasheets, only includes MEMS microphones, ADCs.

4.4. Future Work - Locating Microphone

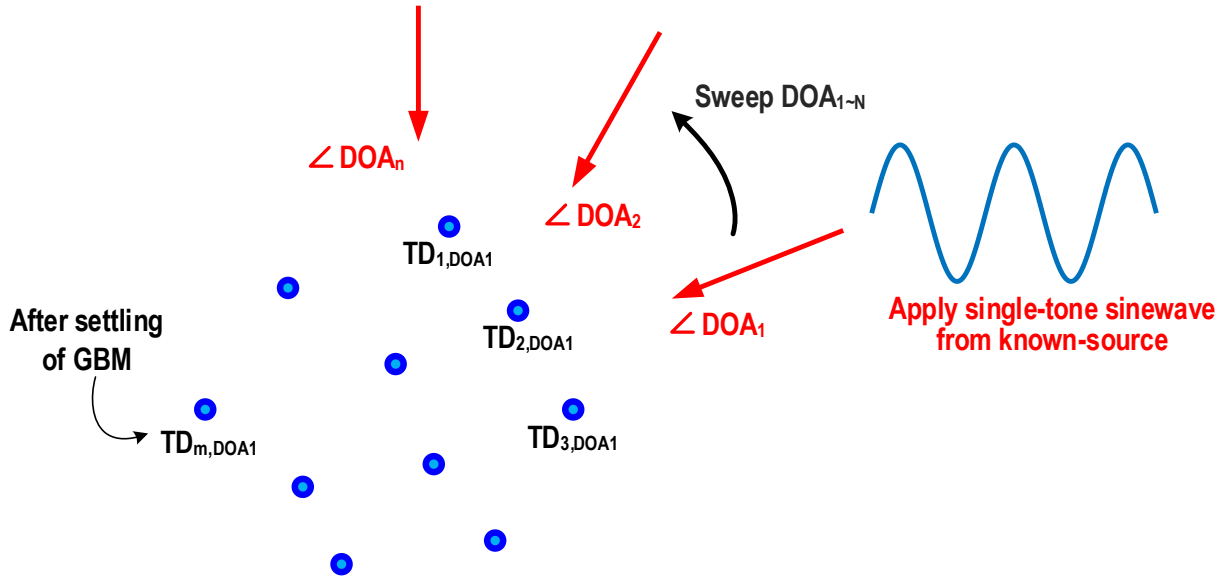


Figure 4-43. The method of locating the microphone using GBM.

We suggest that the proposed GBM algorithm find the location of microphones (or any other sensors) with an external sinewave source, as shown in Figure 4-43. Assume that there are multiple microphones with unknown locations. Then, we apply a single-tone sinewave from the known external source and sweep its DOA. For example, we can get a set of $TD_{1,DOA1}$ to $TD_{m-1,DOA1}$ after convergence of GBM with DOA_1. With enough sweep of DOA, one can use these delays to estimate the microphone array reversely.

The advantage of this method is that it can locate the microphones without extra equipment, such as a camera or ruler. Also, the GBM is very efficient in finding the optimum aligning delay because it does not require excessive parameter sweeps. For example, if there are 100 microphones, one way to find the optimum delay is to adjust the microphone one by one. However, GBM can align 100 microphones at once with some iterations.

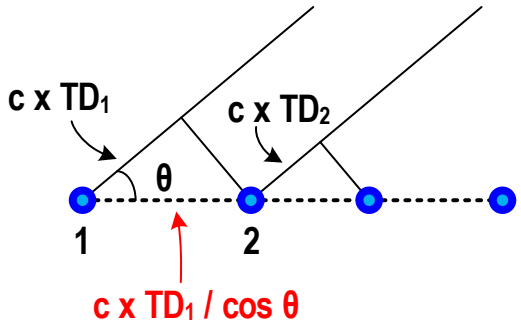


Figure 4-44. Example of locating microphones of a simple linear array.

The simplest example would be a linear microphone array with unknown spacing. In this case, the required number of DOA is just one. After convergence of GBM, the distance between two microphones can be calculated as shown in Figure 4-44.

Chapter 5. Conclusion

This thesis presents three prototype acoustic beamformers with a high SNR ADC array and feature extractor for a complete ASR frontend. In all three, we utilize the direct output of the SDMs before decimation to take advantage of fine delay resolution. In addition, we suggest solutions to deal with wideband speech input and power-hungry hardware.

The prototype in Chapter 2 verifies the effectiveness of the bitstream process regarding steering accuracy. Furthermore, to deal with wideband signals, it realizes CDB by using selective frequency-selective beamforming. However, the fixed DAS has limitations in varying noise situations.

We propose an adaptive beamformer in Chapter 3 to suppress varying noise. Again, the prototype also uses bitstream signals for accurate steering. The RGSC beamformer is effective in varying noise suppression. Also, we present hardware sharing and DSP clock optimization to reduce area and power consumption. The prototype improves speech recognition accuracy in noisy conditions from 64% to 90% using a DNN trained with noisy speech. However, the complicated calculations of the BM hinder the adaptation speed and consume considerable power. An advantage of RGSC is that it separates the target signal and noise – we take advantage of this in the multi-mode beamformer in Chapter 4.

Chapter 4 presents a new beamforming algorithm (GABF) with a four-channel multi-mode ADC array. We focus on two points: 1) multi-mode ADC and beamformer to optimize the power consumption without performance degradation, 2) new beamforming algorithm that solves the conventional RGSC problem. The multi-mode ADC can operate in CTNSSAR (80dBA/12μW),

NSSAR (65dBA/5.8μW), and SAR (40dBA/1.5μW) modes. The system controls the ADC mode depending on the target signal power. Also, the prototype operates the beamformer in two modes by taking advantage of the RGSC structure (separate target and noise path) to save power when the noise level is low. On the other hand, the newly proposed GABF utilizes the direct ADC output with fine delay resolution. Compared to the conventional BM from Chapter 3, the improved design tracks the DOA of the target signal, reduces signal distortion, and reduces power consumption. The 40nm CMOS prototype occupies 0.93mm^2 and consumes 157μW in high-performance mode. It improves KWS accuracy from 54% to 83% under word interference using a DNN trained with noisy speech.

Bibliography

[1] O. Abdel-Hamid, A. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional Neural Networks for Speech Recognition," *IEEE/ACM Trans. Audio, Speech, Language Processing*, vol. 22, no. 10, Oct. 2014.

[2] Available: https://www.globenewswire.com/en/news-release/2021/05/24/2234471/28124/en/Global-Smart-Speakers-Markets-Report-2021-Amazon-Alexa-Google-Assistant-Siri-Cortana-Others-Market-is-Expected-to-Reach-17-85-Billion-in-2025-at-a-CAGR-of-26.html

[3] Y. Zhang, N. Suda, L. Lai, and V. Chandra, "Hello edge: Keyword spotting on microcontrollers," arXiv preprint arXiv:1711.07128 (2017).

[4] Available: https://www.linkedin.com/pulse/how-well-does-speech-recognition-cope-noise-chris-rowen/

[5] Available: https://www.briandorey.com/post/echo-dot-3rd-gen-smart-speaker-teardown

[6] T. Wang, Y. Lin, and C. Liu, "A 0.022 mm^2 98.5 dB SNDR Hybrid Audio ΔΣ Modulator With Digital ELD Compensation in 28 nm CMOS," *IEEE Journal of Solid-State Circuits* vol. 50, no. 11, pp. 2655-2664, 2015.

[7] "Facts About Speech Intelligibility," Available: https://www.dpamicrophones.com/mic-university/facts-about-speech-intelligibility

[8] C. Marro, Y. Mahieux, and K. U. Simmer, "Analysis of noise reduction and dereverberation techniques based on microphone arrays with postfiltering," *IEEE Transactions on Speech and Audio Processing* vol. 6, no. 3, pp. 240-259, May 1998.

[9] C. D. Berti, P. Malcovati, L. Crespi, and A. Baschirotto, "A 106 dB A-Weighted DR Low-Power Continuous-Time ΔΣ Modulator for MEMS Microphones," *IEEE Journal of Solid-State Circuits*, vol. 51, no. 7, pp. 1607-1618, Jul. 2016.

[10] S. S. Priyanka, "A review on adaptive beamforming techniques for speech enhancement," in *IEEE Innovations in Power and Advanced Computing Technologies (i-PACT)*, 2017.

[11] G. W. Elko, and A. N. Pong. "A simple adaptive first-order differential microphone," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, Oct. 1995.

[12] S. Lee, T. Kang, J. Bell, M. R. Haghighat, A. J. Martinez, and M. P. Flynn, "An 8-Element Frequency-Selective Acoustic Beamformer and Bitstream Feature Extractor with 60 Mel-Frequency Energy Features Enabling 95% Speech Recognition Accuracy," in *IEEE Symp. VLSI Circuits*, 2020.

[13] S. Lee, T. Kang, J. Bell, M. Haghighat, A. Martinez, and M. P. Flynn. "An Eight-Element Frequency-Selective Acoustic Beamformer and Bitstream Feature Extractor," *IEEE JSSC*, Aug. 2021.

[14] M. Kajala, and M. Hamalainen, "Filter-and-sum beamformer with adjustable filter characteristics," in *IEEE Proc. ICASSP*, pp. 2917-2920, Jul. 2001.

[15] H. Chen, W. Ser, and Z. L. Yu, "Optimal design of nearfield wideband beamformers robust against errors in microphone array characteristics," *IEEE Transactions on Circuits and Systems I*: Regular Papers, vol. 54, no. 9, pp. 1950-1959, Sep. 2007.

[16] O. L. Frost, "An algorithm for linearly constrained adaptive array processing," *Proceedings of the IEEE* vol. 60, no. 8, pp. 926-935, 1972.

[17] O. Hoshuyama, A. Sugiyama, and A. Hirano. "A robust adaptive beamformer for microphone arrays with a blocking matrix using constrained adaptive filters." *IEEE Trans. Signal Processing*, vol. 47, pp. 2677-2684, Oct. 1999.

[18] T Kang, S. Lee, M. Haghigat, D. Abramson, and M. Flynn, "A 650µW 4-channel 83dBA-SNDR Speech Recognition Front-End with Adaptive Beamforming and Feature Extraction," in *IEEE CICC*, Apr. 2021.

[19] T Kang, S. Lee, M. Haghigat, D. Abramson, and M. Flynn, "A 650µW 4-channel 83dBA-SNDR Speech Recognition Front-End with Adaptive Beamforming and Feature Extraction," *IEEE Solid-State Circuits Letters (SSCL)*, Sep. 2021.

[20] L. J. Griffiths, and C. W. Jim, "An alternative approach to linearly constrained adaptive beamforming," *IEEE Transactions on antennas and propagation* vol. 30, no. 1, Jan. 1982.

[21] T. Yoshioka, N. Ito, M. Delcroix, A. Ogawa, K. Kinoshita, M. Fujimoto C. Yu, W. J. Fabian, M. Espi, T. Higuchi, S. Araki, and T. Nakatani, "The NTT CHiME-3 system: Advances in speech enhancement and recognition for mobile multi-microphone devices," in *IEEE ASRU*, Dec. 2015.

[22] T. Higuchi, N. Ito, T. Yoshioka, and T. Nakatani, "Robust MVDR beamforming using time-frequency masks for online/offline ASR in noise," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[23] T. N. Sainath, R. J. Weiss, A. Senior, K. W. Wilson, and O. Vinyals, "Learning the speech front-end with raw waveform CLDNNs," 2015.

[24] T. N. Sainath, R. J. Weiss, K. W. Wilson, A. Narayanan, M. Bacchiani, and A. Senior, "Speaker location and microphone spacing invariant acoustic modeling from raw multichannel

waveforms," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Dec. 2015.

[25] B. Li, T. N. Sainath, R. J. Weiss, K. W. Wilson, and M. Bacchiani, "Neural network adaptive beamforming for robust multichannel speech recognition," *Proc. Interspeech, ISCA*, 2016.

[26] X. Xiao, S. Watanabe, H. Erdogan, L. Lu, J. Hershey, M. L. Seltzer, G. Chen, Y. Zhang, M. Mandel, and D. Yu, "Deep beamforming networks for multi-channel speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[27] T. N. Sainath, R. J. Weiss, K. W. Wilson, B. Li, A. Narayanan, E. Variani, M. Bacchiani, I. Shafran, A. Senior, K. Chin, A. Misra, and C. Kim, "Multichannel signal processing with deep neural networks for automatic speech recognition," *IEEE/ACM Trans. Audio, Speech, Language Processing (TASLP)*, vol. 25, no. 5, pp. 965–979, May 2017.

[28] B. Li, T. N. Sainath, A. Narayanan, J. Caroselli, M. Bacchiani, A. Misra, I. Shafran et al., "Acoustic Modeling for Google Home," in *Interspeech*, pp. 399-403, 2017.

[29] S. Fu, Y. Tsao, X. Lu, and H. Kawai, "Raw waveform-based speech enhancement by fully convolutional networks," in *IEEE Proc. of APSIPA ASC*, Dec. 2017.

[30] Qian, Kaizhi, Y. Zhang, S. Chang, X. Yang, D. Florencio, and M. Hasegawa-Johnson, "Deep learning based speech beamforming." *in IEEE ICASSP*, Apr. 2018.

[31] W. Minhua, K. Kumatani, S. Sundaram, N. Ström, and B. Hoffmeister, "Frequency domain multi-channel acoustic modeling for distant speech recognition," in *IEEE ICASSP*, Apr. 2019.

[32] T Kang, S. Lee, S. Song, and M. P. Flynn, "A Multi-mode 157μW 4-channel 80dBA-SNDR Speech Recognition Frontend with Self-DOA Correction Adaptive Beamformer," in *IEEE ISSCC*, Feb. 2022.

[33] M. Price, J. Glass, and A. P. Chandrakasan, "A low-power speech recognizer and voice activity detector using deep neural networks," *IEEE JSSC*, vol. 53, no. 1, pp. 66-75, 2017.

[34] S. Oh, M. Cho, Z. Shi, J. Lim, Y. Kim, S. Jeong, Y. Chen et al., "An acoustic signal processing chip with 142-nW voice activity detection using mixer-based sequential frequency scanning and neural network classification," *IEEE JSSC*, vol. 54, no. 11, pp. 3005-3016, Sep. 2019.

[35] M. Almekkawy, J. Xu, and M. Chirala, "An optimized ultrasound digital beamformer with dynamic focusing implemented on FPGA," in *IEEE International Conference on Engineering in Medicine and Biology Society (EMBC)*, Nov. 2014.

[36] B. Starkoff "Ultrasound physical principles in today's technology," Australasian Society for Ultrasound in Medicine, Dec. 2015, [Online].
Available: https://onlinelibrary.wiley.com/doi/full/10.1002/j.2205-0140.2014.tb00086.x

[37] S. Freeman, M. Quick, M. Morin, R. Anderson, C. Desilets, and T. Linnenbrink, "Delta-sigma oversampled ultrasound beamformer with dynamic delays," *IEEE Trans. Ultrason., Ferroelect., Freq. Control*, vol. 46, no. 2, pp. 320-332, Mar. 1999.

[38] M. Chen, A. Perez, S. Kothapalli, P. Cathelin, A. Cathelin, and S. Gambhir, "A pixel pitch-matched ultrasound receiver for 3-D photoacoustic imaging with integrated delta-sigma beamformer in 28-nm UTBB FD-SOI," *IEEE JSSC*, vol. 52, no. 11, pp. 2843-2856, Nov. 2017.

[39] P. Warden, "Speech commands: A public dataset for single-word speech recognition," Dataset Version 0.01, 2017. [Online]. Available:
http://download.tensorflow.org/data/speech_commands_v0.01.tar.gz

[40] H. Cox, R. M. Zeskind, and M. M. Owen, "Robust adaptive beamforming," *IEEE Trans. Acoustic, Speech, and Signal Processing*, vol. 35, pp. 1365–1376, Oct. 1987.

[41] O. Hoshuyama, B. Begasse, A. Sugiyama, and A. Hirano, "A real time robust adaptive microphone array controlled by an SNR estimate," in *IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, May 1998.

[42] K. K. Paliwal, J. G. Lyons, and K. K. Wójcicki, "Preference for 20-40 ms window duration in speech analysis," in *Proc. Int. Conf. Signal Process. Comm. (ICSPCS)*, Dec. 2010.

[43] "Introduction to Filter Designer," MathWorks, Available: https://www.mathworks.com/help/signal/ug/introduction-to-filter-designer.html

[44] M. P. Donadio, "CIC Filter Introduction," Available: http://home.mit.bme.hu/~kollar/papers/cic.pdf

[45] S. Pava, R. Schreier, G. Temes, "Understanding Delta-Sigma Converters," 2nd ed., Wiley, 2017.

[46] "Help Center: weightingFilter," MathWorks, Available: https://www.mathworks.com/help/audio/ref/weightingfilter-system-object.html

[47] "Speech Command Recognition Using Deep Learning," Mathworks. Natick, MA, USA. 2021. [Online]. Available: https://www.mathworks.com/help/deeplearning/ug/deep-learning-speech-recognition.html

[48] H. Liu, Y. Lin, Y. Lee, C. Lee, and C. Yang, "A 98.6μW acoustic signal processor for fully-implantable cochlear implants," *in IEEE Int. Symp. on VLSI Design, Automation and Test (VLSI-DAT)*, Hsinchu, 2016.

[49] Wang, Dewei, et al. "A Background-Noise and Process-Variation-Tolerant 109nW Acoustic Feature Extractor Based on Spike-Domain Divisive-Energy Normalization for an Always-On Keyword Spotting Device," in *IEEE ISSCC*, Feb. 2021.

[50] K. Sreenivasa Rao, "*Appendix.A.3*," in Speech Recognition Using Articulatory and Excitation Source Features, Springer, 2016.

[51] A. Raychowdhury, C. Tokunaga, W. Beltman, M. Deisher, J. W. Tschanz, and V. De, "A 2.3 nJ/frame voice activity detector-based audio frontend for context-aware system-on-chip applications in 32-nm CMOS," *IEEE JSSC*, May 2013.

[52] “fdesign.lowpass,” [Online]. Available: https://www.mathworks.com/help/dsp/ref/fdesign.lowpass.html

[53] Obata, Koji, et al. "A 97.99 dB SNDR, 2 kHz BW, 37.1 μW noise-shaping SAR ADC with dynamic element matching and modulation dither effect." IEEE VLSI-Circuits, 2016.

[54] D. V. Complernolle, W. Ma, F. Xie, and M. V. Diest, "Speech recognition in noisy environments with the aid of microphone arrays," Speech Communication, vol. 9, no. 5-6, pp. 433-442, Dec. 1990.