

Research Article

Design Exploration and Performance Strategies towards Power-Efficient FPGA-Based Architectures for Sound Source Localization

Bruno da Silva^{1,2}, **Laurent Segers**¹, **An Braeken**¹, **Kris Steenhaut**^{1,2},
and **Abdellah Touhafi**^{1,2}

¹INDI Department, Vrije Universiteit Brussel, Brussels, Belgium

²ETRO Department, Vrije Universiteit Brussel, Brussels, Belgium

Correspondence should be addressed to Bruno da Silva; bruno.da.silva@vub.be

Received 24 May 2019; Accepted 26 July 2019; Published 15 September 2019

Academic Editor: Tomasz Wandowski

Copyright © 2019 Bruno da Silva et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Many applications rely on MEMS microphone arrays for locating sound sources prior to their execution. Those applications not only are executed under real-time constraints but also are often embedded on low-power devices. These environments become challenging when increasing the number of microphones or requiring dynamic responses. Field-Programmable Gate Arrays (FPGAs) are usually chosen due to their flexibility and computational power. This work intends to guide the design of reconfigurable acoustic beamforming architectures, which are not only able to accurately determine the sound Direction-Of-Arrival (DoA) but also capable to satisfy the most demanding applications in terms of power efficiency. Design considerations of the required operations performing the sound location are discussed and analysed in order to facilitate the elaboration of reconfigurable acoustic beamforming architectures. Performance strategies are proposed and evaluated based on the characteristics of the presented architecture. This power-efficient architecture is compared to a different architecture prioritizing performance in order to reveal the unavoidable design trade-offs.

1. Introduction

Audio streaming applications involve multiple signal processing operations performed in streams of audio signals and, often, on resource and power-limited embedded devices. Many applications demand the processing in parallel of multiple input streams of audio signals while requiring a real-time response. This is the case of microphone arrays, which are nowadays used in many acoustic applications such as hearing aids [1], biometrical systems [2–4], or speech enhancement [5, 6]. Many of these acoustic applications demand an accurate and fast localization of the sound source prior to any operation [7]. Arrays of microphones are used for acoustic sensing to increase the Signal-to-Noise Ratio (SNR) by combining the input signals from the microphones while steering the microphone's response in a desired direction using acoustic beamforming techniques. Such beamforming techniques involve compute-intensive operations which must be optimized, especially when facing real-time constraints.

FPGAs present valuable features which make them interesting computational units to embedded acoustic beamformers. Firstly, a full customization of an architecture enables multisensor real-time systems. Such embedded systems demand low latency, which cannot be achieved on general-purpose CPUs (e.g., microprocessors). Secondly, FPGAs provide reprogrammable circuitries which can become very power efficient thanks to a high-level architecture customization. Although the amount of the programmable logic resources of low-end FPGAs used in embedded systems is relatively low, streaming applications such as sound locators based on acoustic beamforming can largely benefit from the FPGA's features. Real-time behavior and power efficiency are priorities for sound locators based on acoustic beamforming since a low latency is demanded to estimate the sound Direction-of-Arrival (DoA) while consuming as low power as possible.

We propose several design considerations and performance strategies to fully exploit the current FPGA's capabilities. On the one hand, design considerations are needed to

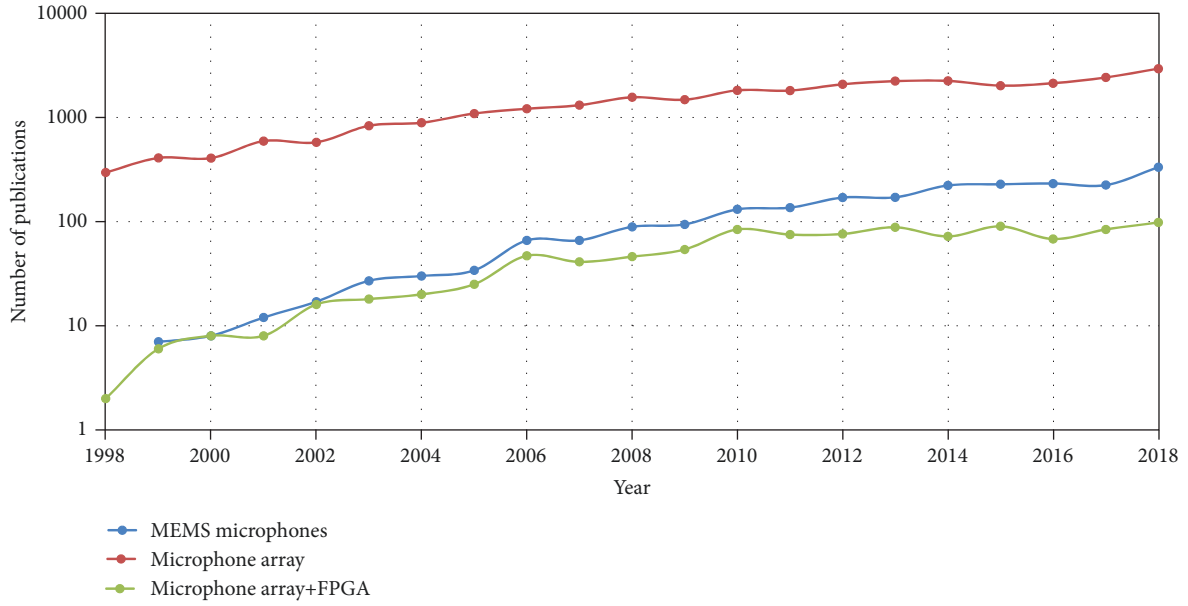


FIGURE 1: Number of publications reported in Google Scholar related to microphone arrays, MEMS microphones, and FPGAs.

properly satisfy the main priority of the acoustic application. Power efficiency is a key feature of the proposed architecture. On the other hand, performance strategies are proposed to accelerate this power-efficient architecture. Each performance strategy firstly considers the architecture's characteristics before exploiting the FPGA's features. As a result, reconfigurable acoustic beamforming architectures are able to operate certain orders of magnitude faster.

The presented work extends the architecture proposed in [8]. Several improvements, such as performance strategies to accelerate the proposed power-efficient architecture, are now added. Moreover, the features of alternative architectures are discussed in detail when describing the required operations for sound source localization and their implementation. The main extensions and new results presented in this work are

- (i) a complete design exploration of FPGA-based architectures using a time-domain Delay-and-Sum beamforming technique for sound source localization is used to identify the more power-efficient architectures
- (ii) performance strategies are proposed to accelerate a power-efficient architecture
- (iii) a detailed comparison between the proposed low-power architecture and the high-performance architectures described in [9, 10] helps to identify the trade-offs when targeting power efficiency

This paper is organized as follows. An overview of related literature is presented in Section 2. A detailed description of the required stages is done in Section 3 in order to properly understand the impact of the architecture's parameters. In Section 4, the metrics used to evaluate acoustic beamforming architectures are described. A power-efficient reconfigurable

acoustic beamforming architecture, which embeds not only a time-domain Delay-and-Sum beamformer but also all the operations to demodulate the Pulse Density Modulation (PDM) signals from the microphones, is proposed and analysed in Section 5. This section exemplifies how performance strategies can fully exploit the architecture's characteristics to accelerate the sound localization. A final comparison with real-time architectures proposed in [9, 10] is done in Section 6 in order to emphasize the existing trade-offs when targeting power efficiency. Finally, some conclusions are drawn in Section 7.

2. Related Work

The interest in microphone arrays has increased in the last decade, partially thanks to recent advances in microelectromechanical systems (MEMS) which have facilitated the integration of microphone arrays in smartphones [11], tablets, or voice assistants, such as Amazon's Alexa [12]. The digital output formats like PDM or Inter-IC Sound (I^2S) currently offered by MEMS microphones facilitate their interface to FPGA-based systems. Figure 1 depicts the number of papers related to microphone arrays, MEMS microphones, and FPGA-based microphone arrays since 1997. The number of publications related to microphone arrays has significantly increased in the last decades (notice the log scale in the number of publications). There is a relation between the evolution of the MEMS technology and the replacement of Digital Signal Processors (DSPs) by FPGAs for audio processing. The majority of the publications related to MEMS microphones or microphone arrays mainly discuss microphone technologies, with around 4% of the overall number of publications describing FPGA-based systems using microphone arrays. We believe that FPGAs are currently underexploited in this area [13].

FPGA-based embedded systems provide enough resources to fully embed acoustic beamformers while presenting power efficiency [8–10]. Such features, however, might not be directly obtained when developing reconfigurable acoustic beamforming architectures. Related literature lacks in design exploration, with few publications discussing architecture's parameters [14, 15] or exploring the full FPGA potential [2, 16].

Fully embedded architectures are, unexpectedly, rare. The available resources and the achievable performance that current FPGAs provide facilitate the signal processing operations required by the PDM demodulation and the beamforming techniques. An example of a fully embedded beamforming-based acoustic system for localization of the dominant sound source is presented in [17, 18]. The authors in [17] propose a FPGA-based system consisting of a microphone array composed of up to 33 MEMS microphones. Their architecture fully embeds the PDM demodulation detailed in [19] together with a Delay-and-Sum beamformer and a Root Mean Square (RMS) detector for the sound source localization.

The authors in [20] also fully embed a beamforming-based acoustic system composed of digital MEMS microphone arrays acting as a node of a Wi-Fi-based Wireless Sensor Network (WSN) for deforestation detection. The proposed architecture performs the beamforming operations before the PDM demodulation and filtering. Instead of implementing individual PDM demodulators for each microphone, the authors propose the execution of the Delay-and-Sum beamforming algorithm over the PDM signals. The output of the Delay-and-Sum, which is no longer a 1-bit PDM signal, is filtered by windowing and processed in the frequency domain. As power consumption is a critical parameter for WSN-related application, their architecture uses an extremely low-power Flash-based FPGA, which allows to only consume 21.8 mW per 8-element microphone array node. A larger version of this microphone array, composed of 16 microphones, is proposed by the author in [21]. Their architecture is migrated to a Xilinx Spartan-6 FPGA due to the additional computational operations, leading to 61.71 mW of power consumption. FPGA-based low-power architectures for WSN nodes to perform sound source localization are, however, not an exception. The authors in [8] propose a multimode architecture implemented on an extremely low-power Flash-based FPGA, achieving a power consumption as low as 34 mW for a 52-element microphone array proposed in [22]. In these architectures, the strategy of beamforming PDM signals has the benefit of saving area and power consumption due to the drastic reduction of the number of filters needed. The architecture's trade-offs, like the real-time capabilities, are, however, not discussed.

Current low-end FPGAs provide enough resources to perform in real-time complex beamforming algorithms involving tens of microphones. Nevertheless, the choice of the architecture is strongly linked to the characteristics and constraints of the target application. Here, a power-efficient architecture is proposed to fully exploit the power-efficient but also resource-constrained FPGAs.

3. Stages of Reconfigurable Architectures for Time-Domain Acoustic Beamforming

Reconfigurable acoustic beamforming architectures share several common components to perform the signal processing operations required to locate sound sources using acoustic beamforming techniques with a MEMS microphone array. The mandatory operations can be grouped in several stages embedding the processing of the acquired data from the MEMS microphone array on the FPGA. Although the microphone array is an external sensing component from the FPGA perspective, its features directly determine some of the architecture's characteristics. Nonetheless, the implementation of reconfigurable acoustic beamforming architectures demands a study and analysis of the impact of the application's parameters. For instance, the sampling frequency (F_s) of the audio input determines the filters' response at the PDM demodulation stage. F_s also affects the beamforming operation, affecting the FPGA resource consumption, which might be critical when targeting small FPGA-based embedded systems.

The impact of the design parameters on the implementation is analysed in this section. Firstly, the required stages in the operations for the audio retrieval, beamforming operations, and the sound localization are detailed. Secondly, a design space of each stage is explored in order to identify the key design parameters and their impact. Such a Design-Space Exploration (DSE) is general enough to obtain power-efficient as well as high-performance reconfigurable architectures such as the one presented in [10]. We start with a short overview of the required stages enabling audio retrieval, beamforming, and sound localization.

Microphone array: digital MEMS microphones, especially PDM MEMS ones, have become popular when building microphone arrays [13]. Besides the multiple advantages of MEMS microphones, some of their features, such as low-power modes, make them interesting candidates for reconfigurable acoustic beamforming architectures. A generic microphone array is used to exemplify how the low-power mode of PDM MEMS microphones can be exploited to construct arrays supporting a variable number of active microphones. Such flexibility demands additional design considerations.

Filter stage: instead of integrating the PDM demodulator in the microphone package, PDM MEMS microphones output a single-bit PDM signal, which needs to be demodulated at the processing side. The PDM demodulation requires additional computations, rather undesirably seen in the relatively low amount of resources available on FPGAs in embedded systems, but it also presents an opportunity to build fully customized acoustic beamformer architectures targeting sound location. The PDM demodulation must also be flexible enough to support dynamic microphone arrays while being power- and resource-efficient. The parameters which determine the required filter response are here identified and used to evaluate multiple designs.

Beamforming stage: its relatively low complexity makes the Delay-and-Sum technique the most popular acoustic beamforming technique. The inherent parallelism in large

TABLE 1: Architecture's parameters used for the analysed reconfigurable acoustic beamforming architectures, the range under evaluation, and their trade-offs.

Parameter	Range	Trade-offs
Number of active microphones (N_{am})	4, 12, 28, 52	Resources, frequency response, directivity, and power
Number of steered orientations (N_o)	4, 8, 16, 32, 64	Execution time, performance, frequency response, and directivity
Number of acoustic samples at the power stage (N_s)	64	Power, execution time, and performance

microphone arrays can be exploited when embedding this type of beamformer on FPGAs. Although the computation of the Delay-and-Sum in the frequency domain is mentioned in the literature, its execution in the time domain is preferred since it avoids the computation of the discrete Fourier transformation, which is a time- and resource-demanding computation. The complex data representation of the data in the frequency domain demands a high bit width or even the use of floating-point representation, leading to multiple Multiply-ACCumulate (MACC) operations to perform the phase shift corrections needed to compensate the difference in path lengths. Instead, the computation of the Delay-and-Sum beamforming technique in the time domain is reduced to the storage of audio samples to synchronize the acquired audio samples for a particular steered orientation. The consumption of the FPGA internal memory, which is used to properly delay the audio samples during the beamforming operation, can be optimized through a judicious choice of design parameters.

Power stage: the direction of the sound source is located by measuring the Sound Relative Power (SRP) per horizontal direction. The SRP obtained by a 360° sweep overview of the surrounding sound field is known as *Polar Steered Response Power* (P-SRP), which provides information about the power response of the array. The P-SRP is only obtained after the audio recovery and the beamforming operation. The accuracy of the DoA based on the P-SRP is determined by parameters such as the number of steered directions. The impact of this parameter is evaluated in the DSE.

The parameters leading to a dynamic response of the sound locator, which can be adapted at runtime thanks to the FPGA's flexibility, are firstly presented together with their unavoidable trade-offs.

3.1. Parameters for a Runtime Dynamic Response. FPGAs present an opportunity to develop dynamic reconfigurable acoustic beamforming architectures, which self-adapt their configuration to satisfy certain criteria. The power consumption, for instance, can be dynamically adjusted at runtime. This dynamism is obtained by exploiting the following architecture parameters at the design stage.

Active microphones: the number of active microphones in the array (N_{am}) directly affects power consumption, frequency response, and performance. The architecture, however, must be designed to support a variable N_{am} . The architecture must be able to selectively deactivate the PDM MEMS microphones of the array, for instance, through the clock signal, and be able to deactivate the

FPGA resources associated with disabled microphones. The following DSE demonstrate how this deactivation can be supported at runtime, without requiring a partial reconfiguration of the FPGA.

Angular resolution: one of the parameters which determine the capability to properly determine the DoA is the angular resolution. The number of steered orientations (N_o) defines the achievable angular resolution when calculating the P-SRP. Similar to N_{am} , this parameter affects the frequency response, the performance, and, indirectly, the power consumption. With these features in mind, the architecture can be designed to support a runtime variable angular resolution as presented in [9, 10].

Sensing time: the sensing time (t_s), a well-known parameter of radio frequency applications, represents the time the receiver is monitoring the surrounding sound field. This parameter is known to increase the robustness against noise [23] and directly influences the probability of proper detection of a sound source. The value of t_s is determined by the number of processed acoustic samples at the power stage (N_s). A higher N_s is needed to detect and to determine the direction of the sound sources under low SNR conditions. Reconfigurable acoustic beamforming architectures can certainly support a variable N_s to adapt at runtime the sensing of the array based on a continuous SNR estimation. Although the proposed architecture must support a variable sensing time at runtime, the evaluation of this parameter is out of the scope of the presented work.

The three parameters are used to provide dynamism to reconfigurable acoustic beamforming architectures. Note that the selection at runtime of the values of N_{am} , N_o , and N_s leads to multiple trade-offs, as already summarized in Table 1. The exact values used for the architecture's analysis are detailed in Table 1.

3.2. Description of the Stages

3.2.1. PDM MEMS Microphone Array. The position of the microphones into the array, known as the array geometry, does affect not only the system's response but also the parameters described in Section 3.1. Moreover, the grouping of microphones in subarrays enables a variable N_{am} and frequency response [9, 10]. This is a topic that has been largely explored ([24, 25] or [26]) and is out of the scope of this work. The microphone array used to evaluate the proposed reconfigurable architectures presents the following characteristics to achieve the desired dynamism:

- (i) The array is composed of PDM MEMS microphones

- (ii) The PDM MEMS microphones support low-power modes (a.k.a. sleep modes)
- (iii) The microphones are grouped in subarrays through a common clock signal

The reference microphone array is composed of 52 digital PDM MEMS microphones like described in [22]. The array geometry consists of four concentric subarrays of 4, 8, 16, and 24 PDM MEMS microphones mounted on a 20 cm circular printed circuit board, depicted in Figure 2. Each concentric subarray has a different radius and number of microphones to facilitate the capture of spatial acoustic information using a beamforming technique. The selection of PDM MEMS microphones is also motivated to the multiple modes that such microphones support. Most of the PDM MEMS microphones offer a low-power mode and drastically reduce their power consumption when the microphones' clock signal is deactivated. This feature allows the construction of microphone arrays composed of multiple subarrays. The response of these microphone arrays can be dynamically modified by individually activating or deactivating subarrays. This distributed geometry can also adapt the architecture's response to different sound sources. For instance, not all subarrays need to be active to detect a particular sound source. The value of N_{am} has a direct impact on the array's output SNR since the SNR increases with N_{am} . In this regard, the computational requirements drastically decrease and the sensor array becomes more power efficient if only a few subarrays are active.

The features of the described microphone array, like the deactivation of the microphones or their group in subarrays, lead to microphone arrays with dynamic response, ideal for high-performance or power-efficient reconfigurable architectures.

3.2.2. Filter Stage. The audio retrieval from PDM MEMS microphones requires certain operations. The first operation to be performed on the FPGA is the PDM demultiplexing since every pair of microphones has its PDM output signal multiplexed in time. The PDM demultiplexing is a mandatory operation to retrieve the individual sampled audio data from each microphone. The incoming data from one of the microphones is sampled at every clock edge. A PDM splitter block, located on the FPGA, demultiplexes the PDM samples.

(1) PDM Demodulators. Figure 3 depicts the internal components of a PDM MEMS microphone. The MEMS transducer converts the input Sound Pressure Level (SPL) to a voltage. This transducer is followed by an impedance converter amplifier, which stabilizes the output voltage of the MEMS for the Sigma-Delta ($\Sigma\Delta$) modulator. The analog signal is digitalized at the ADC and converted into a single-bit PDM signal by a fourth-order $\Sigma\Delta$ modulator running at a high oversampling rate. PDM is a type of modulation used to represent analog signals in the digital domain, where the relative density of the pulses corresponds to the analog signal's amplitude. The $\Sigma\Delta$ modulator reduces

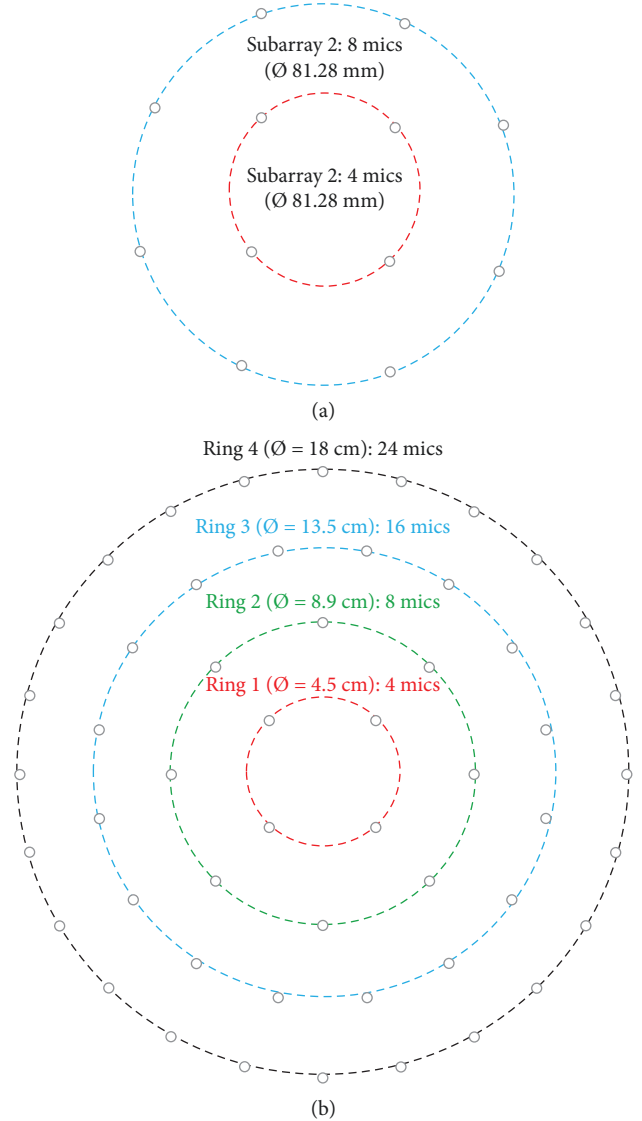


FIGURE 2: Examples of microphone arrays composed of two subarrays ((a) [39]) or four subarrays ((b) [22]). The reference microphone array used to evaluate the proposed power-efficient architecture for the sound source localization is the one described in [22].

the added noise in the audio frequency spectrum by shifting it to higher frequency ranges. This undesirable high-frequency noise needs to be removed when recovering the original audio signal.

Digital MEMS microphones usually operate at a clock frequency ranging from 1 MHz to 3.072 MHz [27] or up to 3.6 MHz [28]. This range of F_s is chosen to oversample the audio signal in order to have sufficient audio quality and to generate the PDM output signal in the $\Sigma\Delta$ modulator. The PDM signal needs not only to be filtered in order to remove the noise but also to be downsampled to convert the audio signal to a Pulse-Code Modulation (PCM) format.

Several examples of PDM demodulators proposed in the literature and incorporated in commercial MEMS

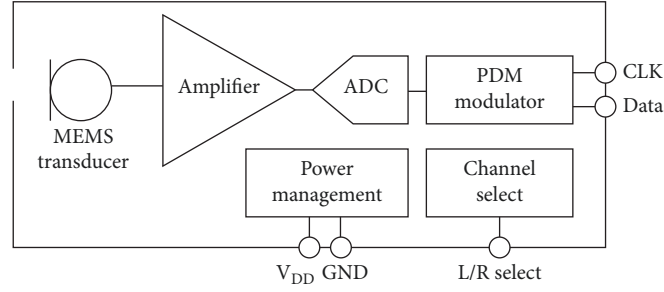


FIGURE 3: Typical digital PDM MEMS microphone block diagram. Source: [40].

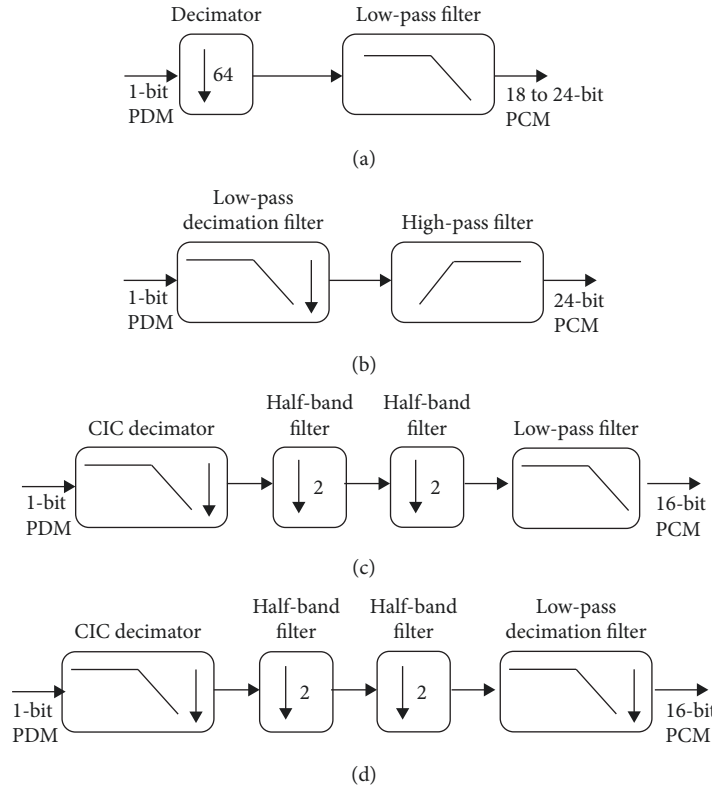


FIGURE 4: Examples of PDM demodulators: (a) is the internal block diagram of the digital I^2S MEMS microphone SPH0645LM4H from Knowles [29]. (b) is the internal block diagram of the digital I^2S MEMS microphone ICS-43434 from TDK InvenSense [30]. (c) is proposed for the Blackfin processor [19]. (d) is proposed in [14].

microphones are depicted in Figure 4. For instance, the PDM demodulators in Figures 4(a) and 4(b) are the block diagrams of the I^2S MEMS microphones [29, 30], respectively. The PDM demodulator in [29] incorporates a decimator to downsample the PDM signal by a factor of 64 and converts the signal to PCM. The remaining high-frequency components in the PCM signal are removed by a low-pass filter. The PDM demodulator in [30] is composed of two cascaded filters acting as a digital band-pass filter. The first one is a low-pass decimator filter which eliminates the high-frequency noise, followed by a high-pass filter, which removes the DC and the low-frequency components. Notice that the decimation factor and the filters' response are fixed in both cases. This fact reduces the DSE since it limits the operational frequency range of the target acoustic application to be a fixed mul-

tipole of the microphones' sampling frequency F_s . For instance, if the PDM demodulator decimates by a fixed factor of 64 like in [29], the microphones must be clocked at $F_s = 3.072$ MHz for a desired output audio at 48 kHz. At that frequency, audio signals up to 24 kHz can be recovered without aliasing according to the Nyquist theorem.

PDM demodulators in Figures 4(c) and 4(d), which are proposed in [19] and in [14], respectively, present a cascade of three different types of filters in a filter chain fashion, that is, a CIC decimation filter followed by two half-band filters with a decimator factor of 2 and a low-pass FIR filter in the final stage. The CIC filters are used to convert the PDM signals in PCM format. This class of linear phase FIR filters, developed by Hogenauer [31, 32], involves only additions and subtractions. It consists of 3 stages: the integrator stage, the decimator or interpolator stage, and the comb section.

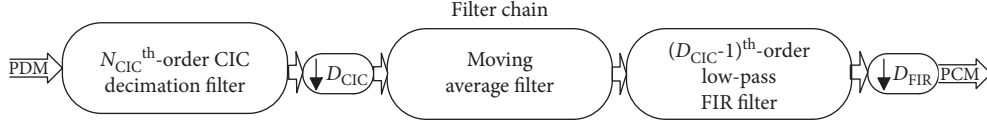


FIGURE 5: The filtering stage consists of single or multiple filter chains performing the PDM demodulation.

PDM input samples are recursively added in the integrator stage while being recursively subtracted with a differential delay in the comb stage. The number of recursive operations in the integrator and comb section determines the order of the filter (N_{CIC}). This order should at least be equal to the order of the $\Sigma\Delta$ converter from the DAC of the microphones. After the CIC filter, the signal growth (G) is proportional to the decimation factor (D_{CIC}) and the differential delay (D) and is exponential to the filter order [32]. CIC decimation filters decimate the signal by D_{CIC} and convert the PDM signal in PCM at the same time. A major drawback of this type of filter is the nonflat frequency response in the desired audio frequency range. To improve the flatness of the frequency response, a CIC filter with a lower decimation factor followed by compensation filters is usually a better choice, as proposed in [19, 32, 33]. The CIC filter is followed by a couple of half-band filters of order N_{HB} with a decimation factor of two. Half-band filters are widely used in multirate signal processing applications. These types of filters that let only half of the frequency band of the input signal present two important characteristics. Firstly, the passband and stopband ripple must be the same. Secondly, the passband-edge and stopband-edge frequencies are equidistant from the half-band frequency $\pi/2$. As a result, the filter's coefficients are symmetrical and every second coefficient is zero. Both characteristics can be exploited for resource savings. The last component is a low-pass compensation FIR filter of order N_{FIR} to remove the high-frequency noise introduced by the ADC conversion process in the microphone. This filter can also be designed to compensate the passband drop usually introduced by CIC filters [32]. Optionally, it can additionally perform a downsampling of the signal being further decimated by a factor of D_{FIR} like that proposed in Figure 4(d).

(2) *Proposed PDM Demodulator.* The analysed filter stage, originally proposed in [10] and in [8], is composed of single or multiple filter chains performing the PDM demodulation. Each filter chain corresponds to several cascaded filters performing a PDM demodulation of the microphone array output signals (Figure 5), simplifying the PDM demodulators in Figures 4(c) and 4(d) by reducing the number of the cascaded filters. Both half-band filters are replaced by a moving average filter, which removes the DC level of the CIC's output signal, improving the dynamic range of the signal entering the low-pass compensation FIR filter. The FIR filter presents a cut-off frequency of F_{max} at a sampling rate of F_s/D_{CIC} , which is the sampling rate obtained after the CIC decimator filter with a decimation factor of D_{CIC} . The stream nature of such architecture enables the generation of an output value from the CIC filter every clock cycle. Due to the decimation factor, only one output value per D_{CIC} input value is propa-

gated to the low-pass FIR filter. Consequently, the FIR filter has D_{CIC} clock cycles to compute each input value. This low-pass FIR filter needs to be designed in a serial fashion to reduce the resource consumption, and its maximum order is also determined by D_{CIC} :

$$N_{FIR} \leq D_{CIC} - 1. \quad (1)$$

Hereby, N_{FIR} is assumed to be equal to its maximum order ($D_{CIC} - 1$) since the order is directly related to the quality of the response of the filter. The overall D_F can be expressed based on the downsampling rate change of each filter

$$D_F = \left\lceil \frac{F_s}{2 \cdot F_{max}} \right\rceil = \left\lceil \frac{F_s}{BW} \right\rceil = D_{CIC} \cdot D_{FIR}, \quad (2)$$

where D_{FIR} is the decimation factor needed for the FIR filter to obtain the minimum bandwidth BW to satisfy the Nyquist theorem for the target F_{max} .

The filter chain depicted in Figure 5 enables dynamic architectures while performing the PDM demodulation. The range of parameters such as F_s and F_{max} depends on the PDM MEMS microphone specifications. For instance, the PDM MEMS microphone ADMP521 from Analog Devices used in [22] operates at a F_s in a range from 1.25 MHz to 3.072 MHz as specified in [27], and its frequency response ranges from 100 Hz to 16 kHz. The specifications of the acoustic application also determine F_{max} , which must be in the range of the supported frequencies of the microphone. Both parameters, F_s and F_{max} , determine the value of D_F and, therefore, the signal rate of each filter. However, not all possible configurations are supported when specifying the low-pass FIR filter's characteristics. For instance, the passband and the stopband, the transition band, or the level of attenuation of the signal out of the passband limit the supported FIR filter's configurations.

(3) *DSE of the PDM Demodulator.* Based on the previous description of the different filters of the filter chain, a DSE can be done to evaluate the supported configurations. The analysis considers an F_{max} ranging from 13 kHz to 16.5 kHz in steps of 125 Hz, and F_s ranges from 1.25 MHz to 3.072 MHz. All possible combinations of D_{CIC} and D_{FIR} are considered on equation (2), based on the D_F obtained for every possible value of F_s and F_{max} . The low-pass FIR filter parameters are N_{FIR} , which is determined by D_{CIC} , and F_{max} as the cut-off frequency. Each possible low-pass FIR filter is generated considering a transition band of 3.5 kHz and

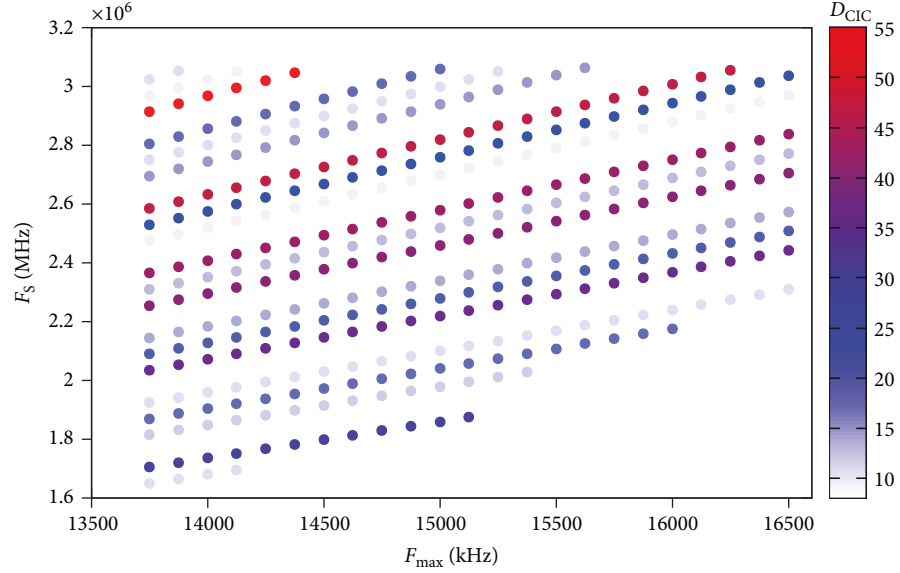


FIGURE 6: Supported values of D_{CIC} based on F_S and F_{max} . The low-pass FIR filter's specifications are detailed in Table 2.

an attenuation of at least 60 dB at the stop band. If the minimum order or the filter is higher than N_{FIR} , the filter is discarded. Furthermore, a minimum order of 8 is defined as the threshold for N_{FIR} . Some values are discarded because D_F is a prime number or N_{FIR} is below 8. The filter's parameters are realistic constraints for low-pass FIR filters.

Figure 6 depicts the values of D_{CIC} for the supported configurations detailed in Table 2. Each low-pass FIR filter is generated and evaluated in MATLAB 2016b. The values of D_{CIC} provide information of D_F and D_{FIR} due to equation (2). Higher values of F_{max} allow higher values of D_{CIC} , which can greatly reduce computational complexity of narrowband low-pass filtering. However, too high values of D_{CIC} lead to such low rates that, although a higher-order low-pass FIR filter is supported, it cannot satisfy the low-pass filtering specifications. Notice how the number of possible solutions decreases when increasing F_{max} . Due to F_S and F_{max} ranges, the values of D_F vary between 38 and 154. Although, as previously explained, many values cannot be considered since they are either prime numbers or the decomposition in factors of D_{CIC} that leads to values below 8. Because higher values of F_{max} lead to low values of D_{CIC} for low F_S , these D_{CIC} values cannot satisfy the specifications of the low-pass FIR filter. High values of D_{CIC} lead to high-order low-pass FIR filters and lower D_{FIR} .

The presented DSE of the filter chain performing the PDM demodulation is general enough to be applied to any of the PDM demodulators depicted in Figure 4. It can be applied to identify the most performing solutions as well as to reduce the resource consumption as discussed in the following section.

3.2.3. Beamforming Stage. Microphone arrays can focus a specific orientation thanks to beamforming techniques. Such techniques amplify the sound coming from the targeted direction while suppressing the sound coming from other

TABLE 2: Parameters used for the design-space analysis.

Definition	Value
F_S	1.25 MHz to 3.072 MHz
F_{max}	13.75 kHz to 16.5 kHz
Steps for F_{max}	125 Hz
Parameters of the low-pass FIR filter	
Minimum order	7
Transition band	4 kHz
Stop band attenuation	60 dB

directions. The time-domain Delay-and-Sum beamforming is a beamforming technique that delays the output signal of each microphone by a specific amount of time before adding all the output signals together. The detection of sound sources is possible by continuously steering in loops of 360° . The number of steered orientations per 360° sweep, N_o , is the angular resolution of the microphone array. Higher angular resolutions demand not only a larger execution time per steering loop but also more FPGA memory resources to store the precomputed delays per orientation.

The beamforming stage performs the time-domain Delay-and-Sum beamforming operation and is composed of a bank of memories, a precomputed table of delays, and several cascaded additions. Although Delay-and-Sum beamforming assumes a fixed number of microphones (N_{Mics}) and a fixed geometry, our scalable solution satisfies those restrictions while offering a flexible geometry [9]. Figure 7 shows our proposed beamforming stage, which is basically composed of FPGA blocks of memory (BRAM) in ring-buffer fashion that properly delays the filtered microphone signal. The delay for a given microphone is determined by its position on the array and on the focus orientation. All possible delay values (Δ) per microphone for each beamed orientation are precomputed, grouped per orientation, and stored in

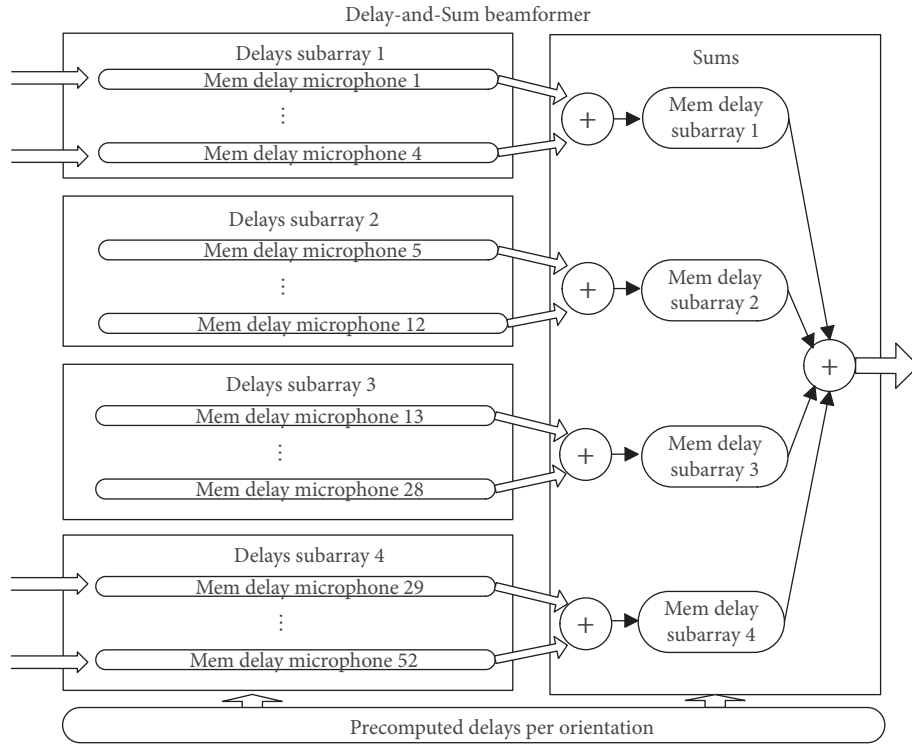


FIGURE 7: Details of the internal structure of the beamforming stage performing the Delay-and-Sum beamforming technique. Note that the delay values are stored in a precomputed table.

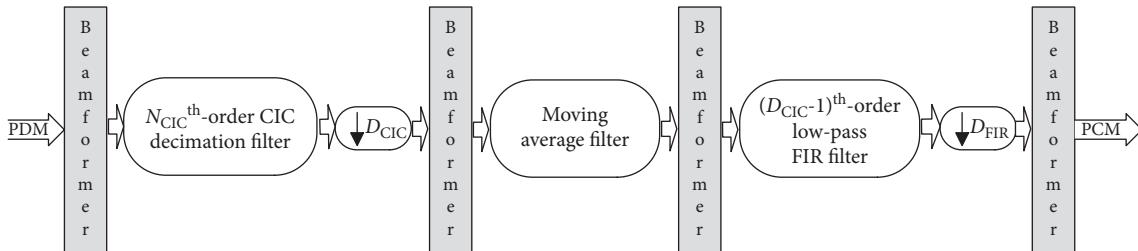


FIGURE 8: Explored locations of the Delay-and-Sum-based beamformer (grey boxes) detailed in Figure 7.

ROMs during compilation time. During execution time, the delay values $\Delta_m(\theta)$ of each microphone m when pointing to a certain orientation θ are obtained from this pre-computed table.

The memory requirements of the beamforming stage are obtained for all the possible locations of the beamforming stage between the components of the filter stage. Figure 8 depicts the potential locations of the Delay-and-Sum-based beamformer. The memory requirements of the beamforming stage based on F_s and on F_{max} are shown in Figure 9. That figure depicts the memory requirements for the supported configurations of the filter chain explored when assuming the FIR filter's characteristics summarized in Table 2. All the discussed characteristics of the filter stage depicted in Figure 6 are evaluated for each possible placement of the Delay-and-Sum-based beamformer. The first possible location is between the microphone array and the CIC filter. The beamforming memory demand linearly increases with F_s . The input signals to be stored are single-bit PDM signals,

which in theory should reduce the need for memory. However, due to high values of Δ_m , thousands of PDM signals need to be stored per microphone. The bit width of the output signals from the CIC filter grows [32], which increases the beamforming memory demands when placing the beamforming stage after the CIC filter. Nevertheless, the signal bit width after the moving average filter and before the low-pass FIR filter can be reduced to 32 bits. Although a lower bit width would not cause significant signal degradation, 32 bits are assumed enough to guarantee the good signal quality for all supported possible microphone array configurations. Due to the audio signal downscaling, low values of Δ_m are obtained when the beamforming stage is located after the low-pass FIR filter, leading to a significant reduction of the memory demands. Detailed analysis of the beamforming memory demands fuels the quest for the most memory efficient architecture.

The memory requirements depicted in Figure 9 have been calculated for a beamforming stage designed to support

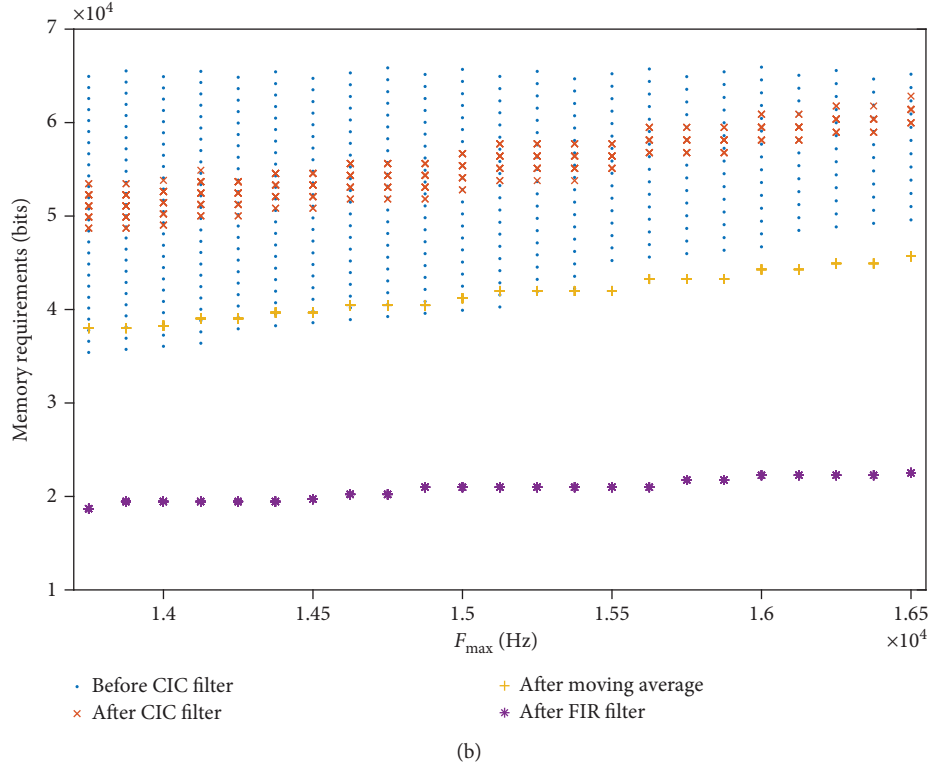
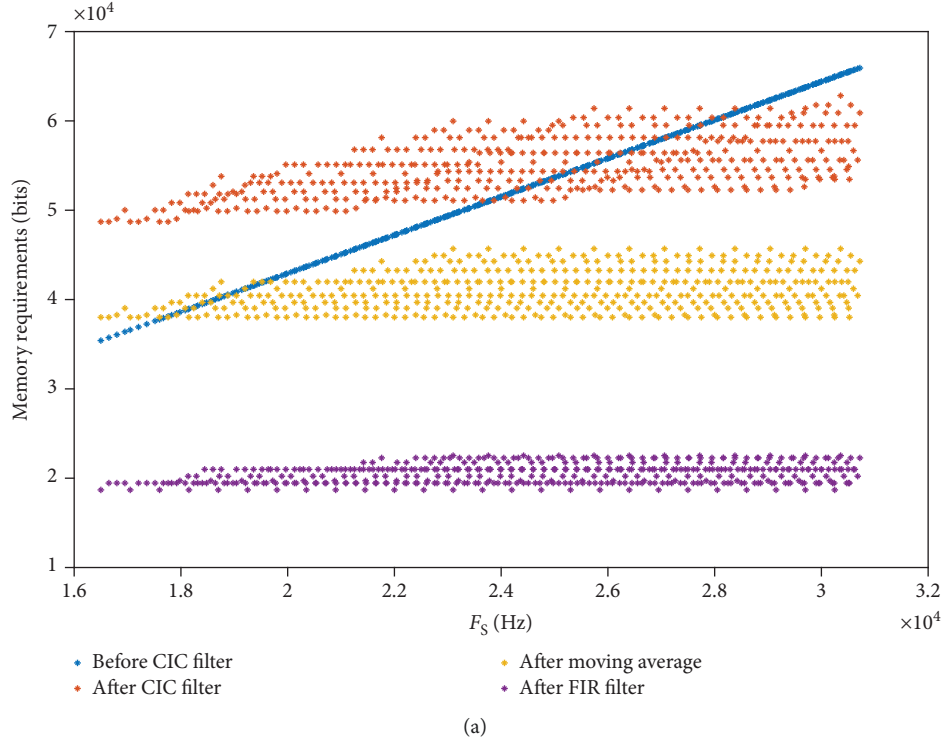


FIGURE 9: Memory consumption based on F_S (a) and F_{\max} (b) for the supported values of D_{CIC} . The memory requirements strongly depend on the position of the beamforming stage in the architecture (Figure 8).

a variable N_{Mics} . The input signals are grouped following their subarray structure. Every microphone m is associated with a memory, which properly delays that particular audio

stream with an amount Δ_m . Each delay memory belonging to a subarray has the same width and length to support all possible orientations. The length is defined by the maximum

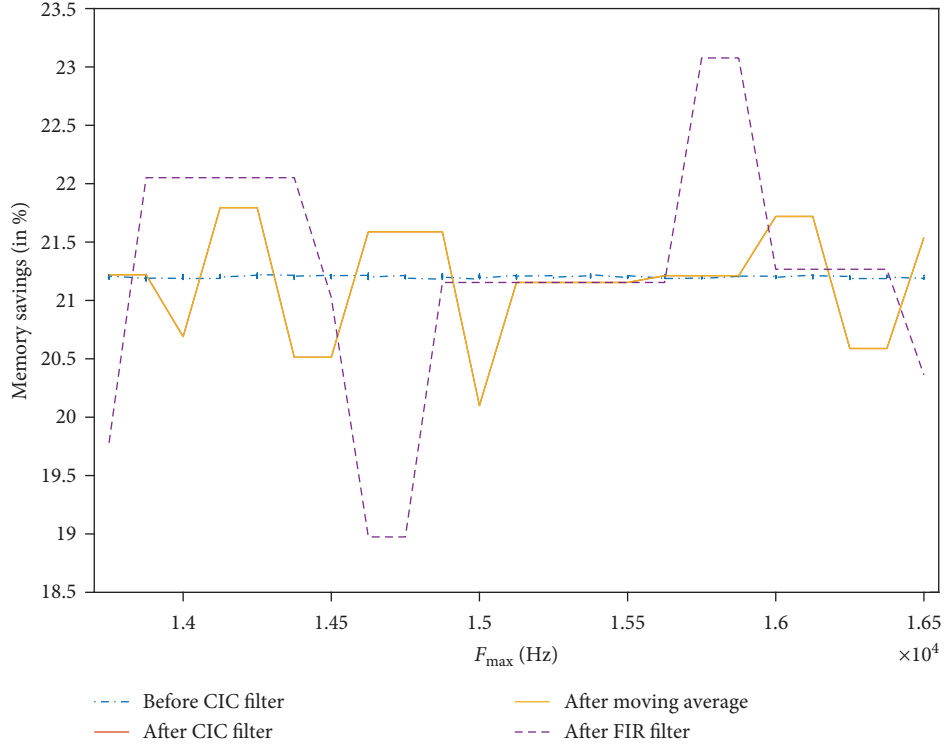


FIGURE 10: Memory savings as a result of decomposing the beamforming stage in subarrays.

delay ($\max(\Delta_{mi})$) of that subarray i , which is determined by the MEMS microphone planar distribution and F_S . All memories associated with the same subarray can be disabled. Therefore, instead of implementing one simple Delay-and-Sum beamformer for a 52-element microphone array, there are four Delay-and-Sum beamforming operations in parallel for the subarrays composed of 4, 8, 16, and 24 microphones. Their sum operation is firstly done locally for each subarray and afterwards between subarrays. The only restriction of this modular beamforming is the synchronization of the outputs in order to have them properly delayed. Therefore, the easiest solution is to delay all the subarrays with the maximum delay ($\max(\max(\Delta_{mi}))$) of all subarrays. Although the output of some subarrays is already properly delayed, additional delays, shown at the Sums part in Figure 7, are inserted to assure the proper delay for each subarray. This is achieved by using the valid output signals of each subarray beamforming, without additional resource cost. Consequently, only the Delay-and-Sum beamforming modulo linked to an active subarray is enabled. The nonactive beamformers are set to zero in order to avoid any negative impact of the beamforming operation.

A side benefit of this modular approach is a reduced memory consumption. Figure 10 shows the memory savings for the supported configurations of the filter chain explored in the previous section. Since each subarray has its ring-buffer memory properly dimensioned to its maximum sample delay, the portion of underused regions of the memories is significantly lower. For the filter chain parameters under evaluation, the memory savings range between 19% and

23%. The variation of the memory savings depends on the placement of the beamforming stage in the architecture. Thus, a mostly constant memory saving of around 21% is possible when the beamforming stage is located between the microphone array and the filter chains. The higher variation occurs when the beamforming stage is located at the end of the filter chains because the memory demands are more sensitive to small differences in the maximum delay values. For instance, whereas in the first case $\max(\Delta)$ rounds to 1048, its value is reduced to 16 when the beamforming stage is located after the filter chain. The modular approach of the beamforming stage does not only increase the flexibility of the architecture by supporting a variable number of microphones at runtime but also represent a significant reduction of the memory requirements.

3.2.4. Power Stage. The Delay-and-Sum beamforming technique allows to obtain the relative sound power of the retrieved audio stream for each steering direction. The computation of the P-SRP in each steering direction provides information about the power response of the array. The power value per steering direction is obtained by accumulating all the individual power values measured for a certain time t_s needed to detect and locate sound sources under low SNR conditions. All the power signals in one steering loop conform the P-SRP. The peaks identified in the P-SRP point to the potential presence of sound sources.

Figure 11 shows the components of the power stage. Once the filtered data has been properly delayed and added, the SRP can be obtained for a particular

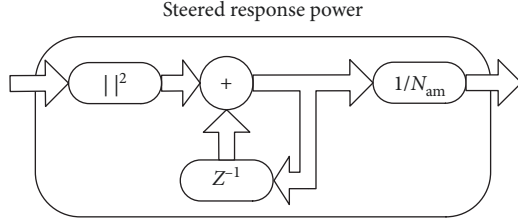


FIGURE 11: The power stage consists of a couple of components to calculate P-SRP, used to estimate the location of the acoustic source.

orientation θ . The P-SRP is obtained after a steering loop, allowing the determination of the sound sources. The sound source is estimated to be located in the direction shown by the peak of the maximum SRP.

3.2.5. Summary. The proposed design considerations and their impact on the architecture are summarized in Table 3. Notice, however, that such design considerations can be individually applied to each stage.

4. Evaluation of Reconfigurable Acoustic Beamforming Architectures

The selection of the design parameters determines the characteristics of the reconfigurable acoustic beamforming architecture. The speed of the architecture, the frequency response, and the accuracy of the sound localization are some of the features used to evaluate and compare designs. The following metrics are used to evaluate the reconfigurable architectures embedding time-domain beamformers for sound location:

- (1) Acoustic response
 - (a) Frequency response
 - (b) Directivity
- (2) Architecture's characteristics
 - (a) Timing and performance
 - (b) Resource and power consumption

4.1. Evaluation of the Acoustic Response. The first two metrics are used to determine the quality of the sound localization and use the array's characteristics to profile the overall response of the selected architecture. The directional power output of a microphone array shows the directional response of the architecture to all sound sources present in a sound field. The lobes of this polar map can then be used to estimate the bearing of nearby sound sources in nondiffuse sound field conditions. The defined P-SRP allows the estimation of the DoA of multiple sound sources under different sound field conditions. The accuracy of its estimation can be determined by the following quality metrics.

Frequency response: the evaluation of the frequency response of the reconfigurable acoustic beamforming architecture for different sound source frequencies is needed. The beam pattern of the microphone array can be represented as a waterfall diagram. Such diagram shows the power output of the sound locator in all directions for all frequencies, which demonstrates how D_p varies with multiple orientations and frequencies. Waterfall diagrams allow the evaluation of the frequency response of different reconfigurable acoustic beamforming architectures for certain beamed orientations. The resolution of the waterfall diagram can be increased by reducing the frequency steps or by increasing the number of steered angles.

Directivity: the P-SRP's lobes are used to estimate the bearing of nearby sound sources in nondiffuse sound field conditions. The capacity of the main lobe to unambiguously point to a specific bearing when considering the scenario of a single sound source determines the architecture's directivity (D_p). This definition of D_p is originally proposed in [34] for broadband signals, where D_p is used as a metric of the quality of the architecture as a sound locator since D_p depends on the main lobe shape and its capacity to unambiguously point to a specific bearing. D_p is a key metric when locating sound sources because it reflects how effectively the architecture discriminates the direction of a sound source. The definition of directivity presented in [34, 35] is adapted for 2D polar coordinates [22] as follows:

$$D_p(\theta, \omega) = \frac{P(\theta, \omega)^2}{(1/2\pi) \int_0^{2\pi} P(\theta, \omega)^2 d\theta}, \quad (3)$$

where $P(\theta, \omega)$ represents the output power of the microphone array when pointing to the sound source's direction θ and $(1/2\pi) \int_0^{2\pi} P(\theta, \omega)^2 d\theta$ is the average output power in all other directions. It can be expressed as the ratio between the area of a circle whose radius is the maximum power of the array and the total area of the power output. Therefore, D_p defines the quality of the sound locator and can be used to specify certain thresholds for the architecture. For instance, if D_p equals 8, the main lobe is eight times lower than the unit circle and offers a trustworthy estimation of a sound source within half a quadrant.

4.2. Evaluation of the Architecture's Characteristics. An evaluation of the reconfigurable acoustic beamforming architecture must cover not only the quality of the sound location but also other implementation-related parameters like the achievable performance and the power consumption.

Time and performance analysis: a proper timing analysis helps to identify performance bottlenecks and to tune the architecture towards lower latency. The time needed by the microphone array to compute P-SRP (t_{P-SRP}) can be determined by decomposing the execution time of the architectural stages. A proper implementation of the stages, and especially of their data flow, can significantly reduce this time. For instance, t_{P-SRP} decreases if the architecture is designed to pipeline the operations of each stage within a steered orientation, enabling the overlapping of the execution

TABLE 3: Summary of the design considerations per stage.

Stage	Design consideration	Effect
Microphone array	PDM MEMS microphones	Deactivation of microphones.
	Subarray decomposition	Power savings.
Filter stage	Moving average filter	Resource savings. Improved dynamic range.
	Serial FIR filter	Resource savings.
Beamforming stage	Subarray decomposition	Power and resource savings.

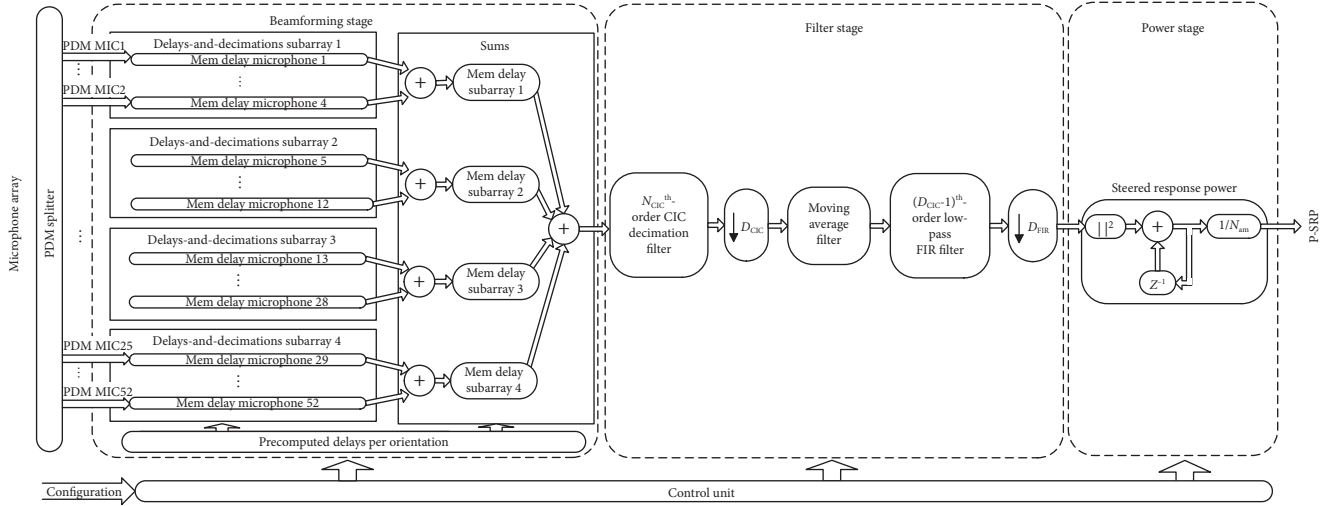


FIGURE 12: Overview of the proposed power-efficient architecture. The Delay-and-Sum beamforming is composed of several memories to properly delay the input signal. Our implementation groups the memories associated to each subarray to disable those memories linked to deactivated microphones. The beamformed input signal is converted to audio in the cascade of filters. The DoA is finally obtained based on SRP obtained per orientation.

of the architecture's components. A detailed analysis of the implementation of each component and its latency provides a good insight in the speed of the system. On the other hand, a performance analysis of a reconfigurable acoustic beamforming architecture gives an idea about what design parameters have a higher performance impact. The performance units can be defined at different levels. The processed audio samples per second reflect the reusability of the acquired data. During the beamforming operation, the same audio sample can be used to calculate the SRP for multiple different orientations. Another performance unit could be the number of beamed orientations per second (Or/s). This type of units better reflects the achievable performance of reconfigurable acoustic beamforming architectures and facilitates the comparison of reconfigurable acoustic beamforming architectures in terms of performance.

Resource and power consumption: further analysis regarding the power or resource consumption of the architecture is needed to satisfy the architecture's target priorities. For instance, the streaming nature of acoustic beamforming applications, with continuous flux of incoming data, needs a large amount of memory to store the intermediate results of the signal processing operations. As analysed in the previous section, a decomposition in subarrays of the beamforming stage reduces the consumption of

internal memory. However, it also affects the power consumption and might finally determine the supported FPGA.

The metrics described above are used in the next section to evaluate a power-efficient architecture. Different performance strategies are proposed to increase the performance of this architecture.

5. Power-Efficient Reconfigurable Architecture

Current low-end FPGAs offer enough resources to embed power-efficient reconfigurable acoustic beamforming architectures such as the one described and analysed in this section. The presented architecture, firstly presented in [8], drastically reduces resource consumption, making it suitable for low-end Flash-based FPGAs. This type of FPGAs presents a power consumption as low as few tens of mW but lacks available resources. Such resource restriction drastically reduces the achievable performance if the architecture's characteristics are not properly exploited. Here, different performance strategies are applied in order to accelerate this architecture, becoming more attractive for time-sensitive applications.

Figure 12 depicts the main components of the power-efficient architecture. The input rate is determined by the microphone's clock and corresponds to F_s . The architecture

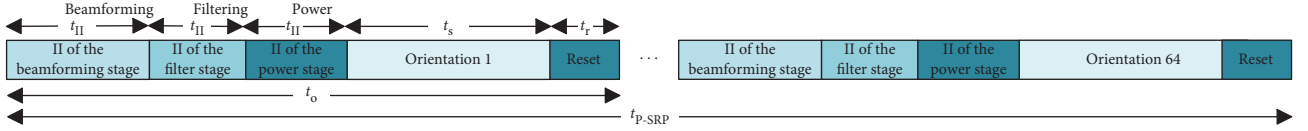


FIGURE 13: Detailed schedule of the operations without any performance strategy.

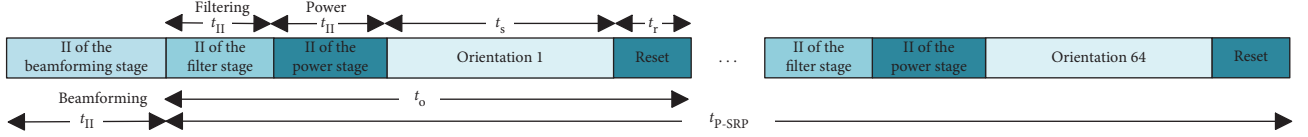


FIGURE 14: Detailed schedule of the operations when continuously beamforming.

is designed to operate in the streaming mode, which guarantees that each component is always computing after an initial latency.

The oversampled PDM signal coming from the microphones is multiplexed per microphone pair, requiring a PDM splitter block to demultiplex the input PDM signal into 2 PDM separate channels. Thus, the PDM streams from each microphone of the array are properly delayed at this stage to perform the Delay-and-Sum beamforming operation. The beamforming stage is followed by the filter stage, where the high-frequency noise is removed and the input signal is downsampled to retrieve the audio signal. Notice that a filter stage is only composed of one filter chain like the one described in Section 3.2.2 instead of N_{Mics} filter chains thanks to placing the beamforming stage before the filter stage. The SRP for the beamed orientation is calculated in the last stage. The lobes of the P-SRP are used to estimate the DoA for the localization of the sound sources.

5.1. Architecture Performance Exploration. The architecture is designed to satisfy power-constraint acoustic beamforming applications. Multiple performance strategies can be applied to increase performance while preserving the power efficiency. Such strategies minimize the timing impact of the signal processing operations of the architecture.

The execution time ($t_{\text{P-SRP}}$) is defined as the time needed to obtain the P-SRP. Each steered orientation involves multiple signal processing operations that can be executed concurrently in a pipelined way. Therefore, the times to filter ($t_{\text{Filtering}}$), to beamform ($t_{\text{Beamforming}}$), and to get the SRP (t_{Power}) are overlapping with the sensing time (t_s). Although most of the latency of each component of the design is hidden when pipelining operations, there are still some cycles, defined as Initiation Interval (II), dedicated to initialize the components. The proposed architecture also demands an additional time to reset the filters (t_r) at the end of the computation of each orientation. The relatively low value of t_r can be neglected because only a few clock cycles are needed to reset the filters. As detailed in Figure 13, $t_{\text{P-SRP}}$ for a certain N_o can be determined by

$$t_{\text{P-SRP}} = N_o \cdot (t_{\text{II}} + t_r + t_s) \approx N_o \cdot (t_{\text{II}} + t_s), \quad (4)$$

where t_{II} corresponds to the sum of the II of the filter stage ($t_{\text{II}}^{\text{Filtering}}$), the II of the beamforming stage ($t_{\text{II}}^{\text{Beamforming}}$), and the II of the SRP stage ($t_{\text{II}}^{\text{Power}}$).

The power-efficient architecture presents several limitations when considering performance strategies. For instance, due to the architecture's characteristics, the strategies proposed in [10] cannot be applied without a significant increment of the resource consumption. Some new performance strategies are here proposed to overcome these limitations.

5.1.1. Continuous Beamforming. The computation of P-SRP considers a reinitialization of the beamformer per beamed orientation. Such initialization can be drastically reduced if the architecture continuously beamforms the acquired data. Due to the fact that the beamforming stage is mainly composed of delay memories, the required data to start the computation of SRP for a new orientation has been already stored when computing the previous orientation. Therefore, a single initialization is needed at the very beginning, as detailed in Figure 14. The value of t_o becomes

$$t_o = t_{\text{II}}^{\text{Filtering}} + t_{\text{II}}^{\text{Power}} + t_r + t_s, \quad (5)$$

since $t_{\text{II}}^{\text{Beamforming}}$ must be only considered when the system starts. In this regard, $t_{\text{P-SRP}}$ becomes

$$t_{\text{P-SRP}} = N_o \cdot t_o \approx N_o \cdot (t_{\text{II}}^{\text{Filtering}} + t_s), \quad (6)$$

where $t_{\text{II}}^{\text{Power}}$ and t_r are neglected.

The performance for using this strategy is defined as follows:

$$\text{Performance} = \frac{1}{t_o} \approx \frac{1}{(t_{\text{II}}^{\text{Filtering}} + t_s)}. \quad (7)$$

5.1.2. Parallel Continuous Time Multiplexing. The single increment of the functional clock frequency beyond F_s is not enough to improve performance improvement. Unfortunately, the simple acceleration by increasing the operational frequency of the filter stage demands the

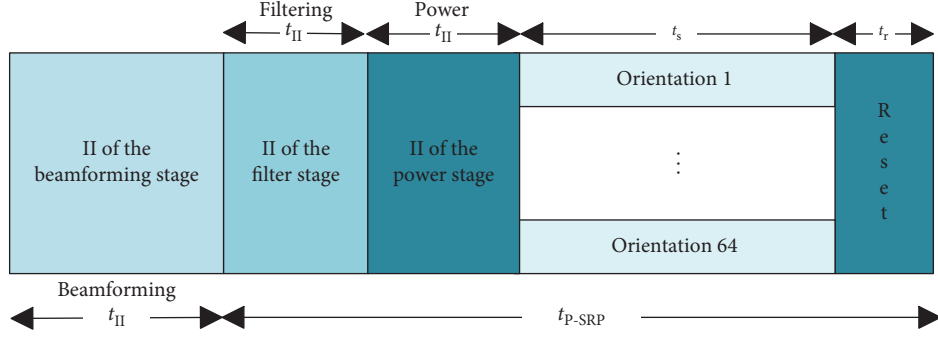
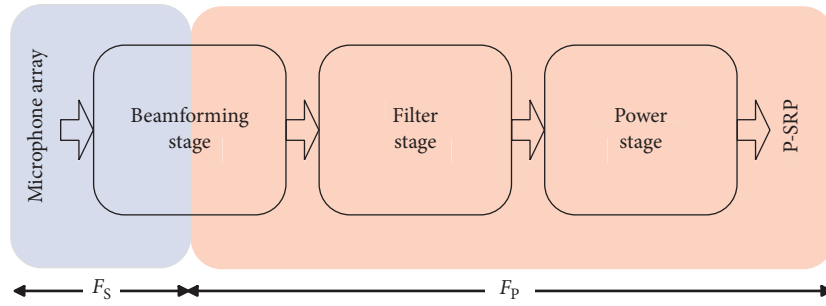


FIGURE 15: Possible schedule of the operations computing in parallel.

FIGURE 16: Clock regions for the time multiplexing of the computation of multiple N_o .

prestorage of all the samples acquired in t_s . Although such storage can take place at the beamforming stage, this component would largely increase its resource consumption to store $N_s \cdot (D_F/F_S)$ samples per microphone. Because of the nature of the filtering operation, the switching between different orientations demands the storage of the intermediate values stored in the multiple taps of the filter structure. Such storage should be applied for each intermediate register of each filter and for each orientation. The impact of this resource overhead would be similar to the cost of replicating each filter chain per orientation. This strategy causes a significant increment of resource consumption due to the fact that the filter stage is located after the beamforming stage. The solution is the replication of the filter stage while increasing its operational frequency to F_P . Figure 15 details how the architecture would perform when multiple filter chains, as many as N_o , are available. Two clock regions are defined since the filter chains must operate at a higher frequency in order to retrieve the beamformed data from the beamforming stage (Figure 16). The incoming data from the microphone array enters the beamforming stage at F_S rate. To process N_o orientations in parallel in one clock cycle at F_S , the beamforming stage needs to generate data at a desired F_P :

$$F_P \geq \frac{F_S \cdot N_{FStages}}{N_B}, \quad (8)$$

where $N_{FStages}$ is the number of filter chains available in the filter stage and N_B is the number of beamformed

values out of the beamforming stage accessible per clock cycle. The value of N_B is defined as

$$N_B = N_{BStages} \cdot M_{ports}, \quad (9)$$

with $N_{BStages}$ the number of beamforming stages if the available resources support more than one and M_{ports} the memory ports of each beamforming memory. For instance, dual-port memories allow 2 readings per memory access, which results in 2 output beamformed values per clock cycle if the sums performed in the beamforming stage are duplicated. The use of dual-port memories is equivalent to duplicating the beamforming stage composed of single-port memories. In both cases, 2 beamformed values can be loaded from each microphone delay memory. This strategy, however, does not exploit the remaining resources to instantiate multiple beamforming stages, and therefore, N_B is assumed to be 1 for this strategy since single-port memories are considered for the beamforming stage.

The value of $N_{FStages}$ not only is determined by the available resources but also depends on N_o and the maximum operational frequency F_{op} . The number of orientations that can be computed in parallel when increasing the operational frequency of the filter stage to F_{op} defines the number of the supported filter stages N_F as

$$N_F = \left\lfloor \frac{F_{op} \cdot N_B}{F_S} \right\rfloor. \quad (10)$$

TABLE 4: Summary of the performance strategies for the described power-efficient architecture.

Strategy	Performance	Time execution of one P-SRP (t_{P-SRP})
None	$\frac{1}{t_{II} + t_s}$	$N_o \cdot (t_{II} + t_s)$
Continuous beamforming	$\frac{1}{t_{II}^{Filtering} + t_s}$	$N_o \cdot (t_{II}^{Filtering} + t_s)$
Parallel continuous time multiplexing	$\frac{N_{FStages}}{t_{II}^{Filtering}} + t_s$ with $N_{FStages} = \min(N_R, N_F)$	$N_o/N_{FStages} \cdot (t_{II}^{Filtering} + t_s)$ with $N_{FStages} = \min(N_o, N_R, N_F)$

Therefore, $N_{FStages}$ becomes

$$N_{FStages} = \min(N_o, N_R, N_F), \quad (11)$$

where N_R is the supported number of filter chains determined by the available resources. Notice that $N_{FStages}$ can be limited to N_o when there is no additional benefit of processing a higher value of N_o , which is determined by the target acoustic application.

With this strategy, the time to compute one orientation is reduced with a factor of $N_{FStages}$:

$$t_o = t_{II}^{Filtering} + t_r + t_s, \quad (12)$$

and the value of t_{P-SRP} becomes

$$t_{P-SRP} = N_o \cdot t_o \approx \frac{N_o}{N_{FStages}} \cdot (t_{II}^{Filtering} + t_s), \quad (13)$$

where t_{II}^{Power} and t_r are neglected. Regarding the achievable performance,

$$\text{Performance} = \frac{N_{FStages}}{(t_{II}^{Filtering} + T_s)}. \quad (14)$$

The cost, however, is an increment of the resources and power consumption. The extra resources are dedicated to the $N_{FStages}$ filter chains or even to additional beamforming stages if $N_{FStages}$ is limited by N_F , in order to fully compute in parallel.

The strategies to improve performance are summarized in Table 4. Notice that their impact is not as significant since the main goal of the power-efficient architecture is to reduce the resource consumption and, as a result, the overall power consumption.

5.2. Experimental Results. The design parameters of the architecture under evaluation are summarized in Table 5. The variation of the target F_{max} and the F_s directly affects the beamforming stage by determining the length of the memories and the filter stage, by determining the decimation factor and the FIR filter order. Moreover, the impact of N_{am} , which changes at runtime thanks to the subarray

TABLE 5: Configuration of the architecture under analysis.

Parameter	Definition	Value
F_s	Sampling frequency	2.08 MHz
F_{min}	Minimum frequency	1 kHz
F_{max}	Maximum frequency	16.250 kHz
BW	Minimum bandwidth to satisfy Nyquist	32.5 kHz
D_F	Decimation factor	64
D_{CIC}	CIC filter decimation factor	32
N_{CIC}	Order of the CIC filter	4
D	CIC differential delay	32
D_{FIR}	FIR filter decimation factor	2
N_{FIR}	Order of the FIR filter	31

distribution, is analysed. Like the evaluation of the previous architecture, P-SRP is obtained from a steering loop composed of 64 orientations. The power-efficient architecture has been evaluated for a Microsemi's SmartFusion2 M2S025 FPGA.

5.2.1. Frequency Response. The frequency response of the microphone array is determined by N_{am} . The experiments cover four configurations with 4, 12, 28, or 52 microphones determined by the number of active subarrays. The waterfall diagram of each configuration is generated in order to analyse the frequency response while locating sound sources. The waterfall diagrams show the power output of the combined subarrays in all directions for all frequencies. The results are calculated with a single sound source placed at 180°. The frequency of the sound source varies between 100 Hz and 15 kHz in steps of 100 Hz. All results are normalized per frequency.

Figure 17 depicts the waterfall diagrams when combining a different number of subarrays. Every waterfall shows a clear distinctive main lobe. However, this lobe dominates the most in case when subarrays 3 and 4 are also capturing sound waves. When only subarray 1 is active, the side lobes affect the capacity of finding the main lobe. The frequency response of the subarrays improves when they are combined since their frequency responses are superposed. Consequently, the combination of the subarrays 1 and 2 reaches a minimum

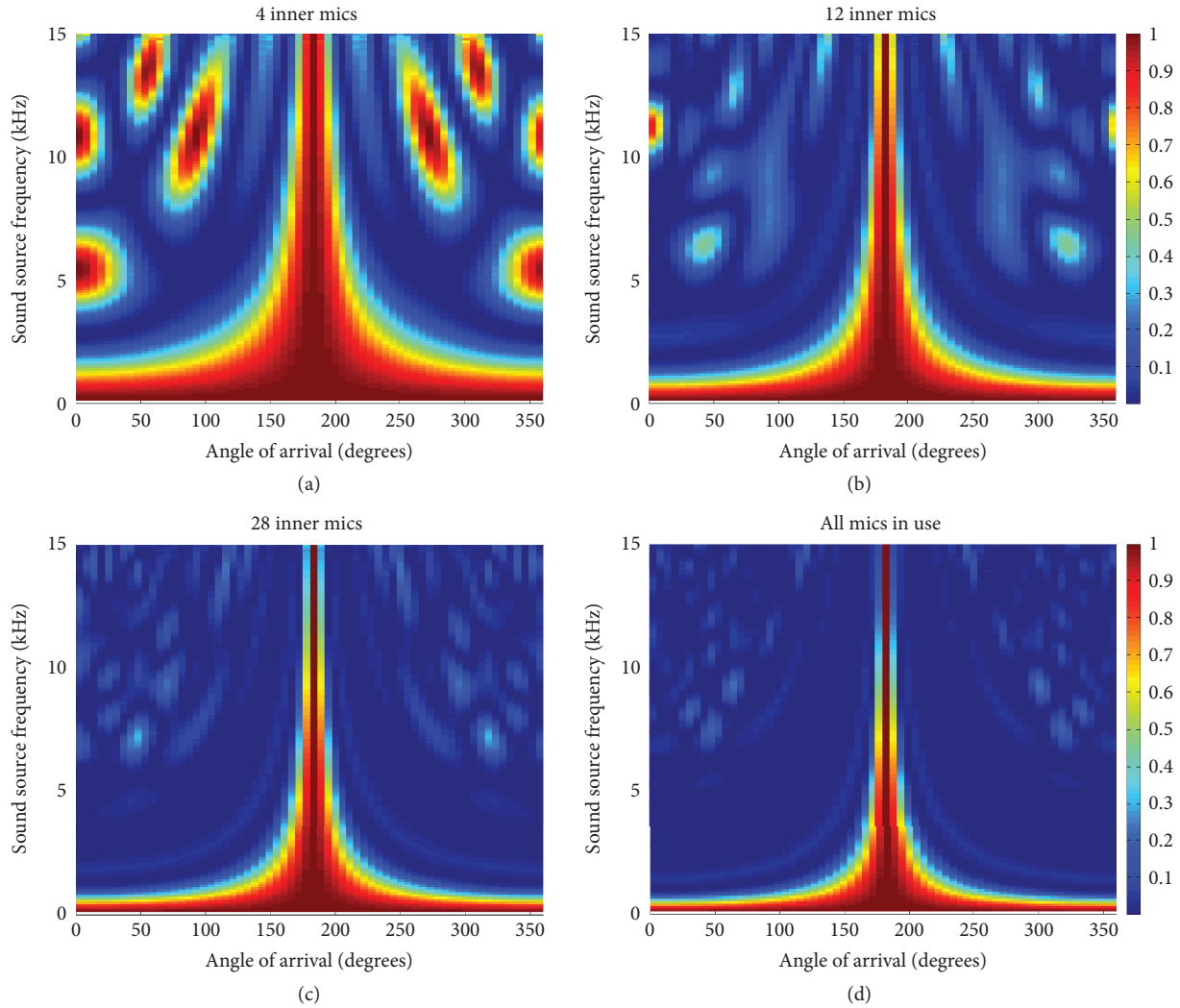


FIGURE 17: Theoretical waterfall diagrams of the power-efficient architecture obtained for 64 orientations. The plots are obtained by enabling only a certain number of subarrays. (a–d) Only the 4 most inner microphones, only the 12 most inner microphones, the 28 most inner microphones, and all microphones.

detectable frequency of 2.4 kHz, whereas the combination of the subarrays 1, 2, and 3 and the combination of all subarrays reach 2.2 kHz and 1.8 kHz, respectively.

5.2.2. Directivity. The standalone waterfall diagrams only provide information about the frequency response but cannot be considered a metric of the quality of the sound source location. Alongside with the waterfalls, D_p is calculated to properly evaluate the quality of the array's response. The evaluation covers a variable N_o and N_{am} . A low angular resolution leads to a lower resolution of the waterfall diagrams, but only the metrics can show the impact. The frequency response of a subarray has a strong variation at the main lobe and, therefore, in D_p . A threshold of 8 for D_p indicates that the main lobe's surface corresponds to maximum half of a quadrant. Figure 18 depicts the evolution of D_p for our frequency range when increasing the angular resolution and when combining subarrays. The angular resolution determines the upper bound D_p converges to, which is defined

in equation (3), and coincides with the number of orientations. Notice that a higher angular resolution does not improve D_p when only the inner subarray is active. The value of N_{am} , on the other hand, determines how fast D_p converges to its upper limit, based on the frequency of the sound source. Thus, a higher value of N_{am} increases D_p for lower sound source frequencies. For instance, in the case that only subarray 1 is used, D_p shows better results only at frequencies beyond 6.9 kHz. This frequency decreases to approximately 1.7 kHz when all microphones are used in the beamforming.

The evaluation of the frequency response of this architecture concludes that the angular resolution determines the quality of the array's response. This is reflected in the D_p , which is clearly limited when reducing the number of orientations. N_{am} determines if at what sound source frequency a certain threshold is achieved. Clearly, a higher value of N_{am} allows the achievement of a better D_p at a lower frequency. A higher number of orientations and active microphones

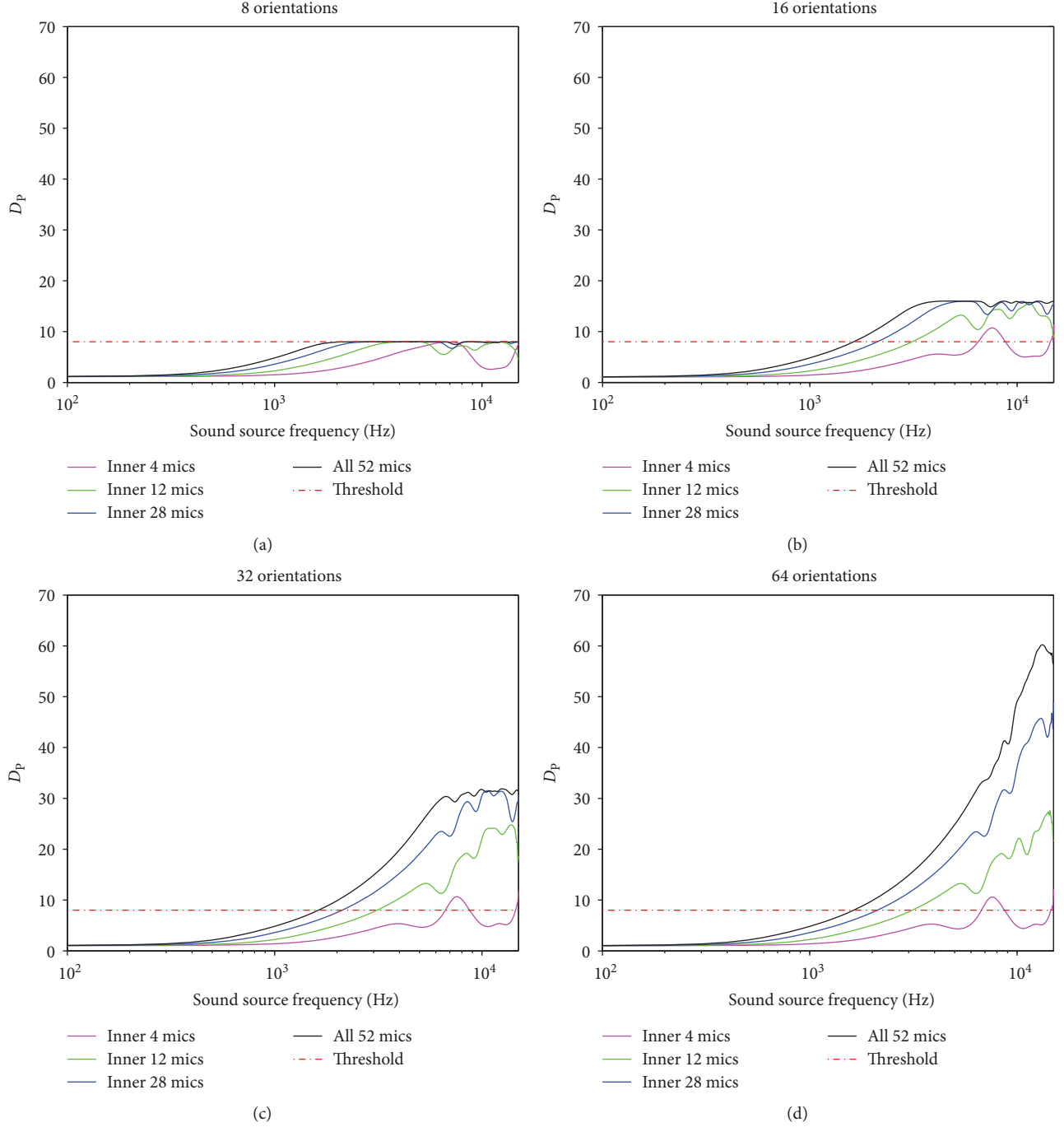


FIGURE 18: Directivities of the power-efficient architecture when considering a variable number of orientations and active microphones. (a–d) The D_p with only 8 orientations up to 64 orientations.

lead to other trade-offs. Whereas the angular resolution will also affect the performance, the N_{am} will determine the power consumption.

5.2.3. Resource Consumption. The relatively low resource requirements of the power-efficient architecture allows the use of small and low-power Flash-based FPGAs. Table 6 summarizes the resource consumption of the evaluated power-efficient architecture. The target SmartFusion2 M2S025 FPGA provides enough resources to allocate one

instantiation of the architecture when fully using all the 52 MEMS microphones of the array. A higher number of subarrays mainly increase the resource consumption of the beam-forming stage. Moreover, the most demanding resources are the dedicated memory blocks, achieving occupancy rates of 76.5% and 90.3% for the two types of memory resources uSRAM (RAM64x18) and LSRAM (RAM1K18), respectively, available in this Microsemi FPGA family [36]. The use of all subarrays demands the use of VHDL attributes to distribute the allocation of the different delay memories of

TABLE 6: Resource consumption after placement and routing when combining the two innermost subarrays and when all microphones are active. Each subarray combination details the resource consumption of the filter and the beamforming stage.

Resources	Available resources	Inner 4 mics			Inner 12 mics			Inner 28 mics			All 52 mics		
		Beamformer	Filters	Total	Beamformer	Filters	Total	Beamformer	Filters	Total	Beamformer	Filters	Total
4LUT	27696	377	839	1626	955	839	2202	2345	839	3592	4303	836	5547
DFP	27696	319	3858	4649	750	3860	5090	1680	3862	6038	2983	3862	7365
Interface logic	27696	144	144	432	432	144	720	1008	144	1296	1872	144	2160
RAM64x18	34	0	2	2	0	2	2	0	2	2	24	2	26
RAM1K18	31	4	0	4	12	0	12	28	0	28	28	0	28
MACC	34	0	2	6	0	2	6	0	2	6	0	2	6

TABLE 7: Power consumption expressed in mW when combining microphone subarrays. The values are obtained from the Libero SoC v.11.8 power report for the FPGA operating at $F_s = 2.08$ MHz and considering the standard mode of the PDM MEMS microphones [38].

Active subarrays	MEMS microphones			Reported on-chip power			Total power
	Active	Inactive	Total	Static	Dynamic	Total	
Inner 4 mics	3.096	1.036	4.132	13.963	3.818	17.781	21.913
Inner 12 mics	9.288	0.864	10.152	15.333	4.831	20.164	30.316
Inner 28 mics	21.672	0.518	22.190	15.933	6.819	22.752	44.942
All 52 mics	40.248	0	40.248	15.943	7.794	23.738	63.986

the beamforming stage between these memory resources, since the Libero 11.8 tool does not automatically allocate the beamforming memories into these memory resources. In fact, the delay values linked to the outer subarray composed of 16 MEMS microphones have to be entirely allocated in uSRAM blocks. The consumption of the logic resources achieves a maximum consumption of 26% of the available D-type Flip-Flops (DFFs). In this regard, some of the performance strategies detailed in Section 5.1 can benefit from the use of the remaining logic resources.

5.2.4. Power Analysis. Flash-based FPGAs like Microsemi's IGLOO2, PolarFire, or SmartFusion2 not only offer the lowest static power consumption, demanding only few tens of mW, but also support an interesting sleep mode called Flash-Freeze. The Flash-Freeze mode is a low-power static mode that preserves the FPGA configuration while reducing the FPGA's power draw to just 1.92 mW for IGLOO2 and SmartFusion2 FPGAs [37].

Table 7 summarizes the reported power consumption. The power consumption of the FPGA design has been obtained by using the Libero SoC 11.8 Power tool, obtaining the dynamic and the static power consumption. Whereas the static power consumption is the power consumed based on the used resources, the dynamic power consumption is determined by the computational usage of the resources and the dynamism of the input data. The static power remains mainly constant since there is no significant increment of the consumption when increasing the number of microphones. The dynamic power consumption, on the contrary, increases since a large number of data must be stored and processed. The overall power consumption of the reconfigurable architecture rounds from 17.8 mW to 23.7 mW, which represents a significant reduction compared to architectures like the high-performance architecture in [9, 10], whose power consumption ranges from 122 mW to 138 mW. Furthermore, the low-power consumption of the FPGA is partially possible thanks to operating at a relatively low frequency (2.08 MHz).

The power consumption analysis must also include the power consumption of the microphone array. For this analysis, the InvenSense ICS-41350 PDM MEMS microphones [38] operating at their standard mode are considered as the microphones composing the array. The power consumption detailed in Table 7 decreases by deactivating the MEMS microphones of the array, which is done by disabling their clock. Thanks to the flexibility of the reconfigurable architecture, N_{am} can be changed at runtime. For the current

measurements, the MEMS microphones are powered with 1.8 V, which represents a power consumption per microphone of $21.6 \mu\text{W}$ and $777 \mu\text{W}$ for the inactive and active microphones, respectively. As a result, the power consumption of the MEMS microphones almost doubles the FPGA's power consumption when all the microphones are active.

5.2.5. Timing Analysis. Table 8 summarizes the parameters of the timing analysis. The value of t_{P-SRP} equals to 174 ms for the analysed architecture when no strategy is applied. Notice that around 22% corresponds to the initialization of the beamforming stage when switching orientation.

The timing results when applying the performance strategies proposed in Section 5.1 are summarized in Table 9. The first strategy reduces the impact of the initialization after transitions between orientations:

$$t_{P-SRP} = N_o \cdot (t_{II}^{\text{Filtering}} + t_s) = 142.1 \text{ ms.} \quad (15)$$

Further acceleration is only possible by increasing the resource consumption while operating the filter stage at a higher frequency. Based on the remaining resources of the SmartFusion2 M2S025, up to 6 filter chains can be allocated in parallel due to the resource consumption of the filter chains detailed in Table 6. The maximum operational frequency of this architecture ranges from 93.11 MHz to 86.92 MHz if only the inner subarray or all subarrays are active, respectively. By operating at 86.92 MHz, and considering single-port memories, up to 43 filter stages can be fetched. The supported number of filter chains, $N_{FStages}$, is obtained from equation (11):

$$N_{FStages} = \min(N_o, N_R, N_F) = \min(64, 6, 43) = 6, \quad (16)$$

where $N_{FStages}$ is limited by the available resources. Therefore,

$$t_{P-SRP} = \frac{N_o}{\min(N_o, N_{FStages})} \cdot (t_{II}^{\text{Filtering}} + t_s) = 23.7 \text{ ms,} \quad (17)$$

and the filter stage needs to operate at least at

$$F_P = F_s \cdot \frac{N_{FStages}}{N_B} = 12.48 \text{ MHz,} \quad (18)$$

in case all the microphones are active.

TABLE 8: Definition of the architecture's parameter involved in the time analysis. N_s is the number of output samples and N_{am} is the number of active microphones.

Parameter	Definition	Equation	Value (cc/MHz)
t_{II}^{CIC}	II of the CIC filter	$2 \cdot N_{CIC} + 1$	$\frac{9}{F_s} = 4.33\mu s$
t_{II}^{DC}	II of the remove DC	$D_{CIC} + 2$	$\frac{34}{F_s} = 16.35\mu s$
t_{II}^{FIR}	II of the FIR filter	$D_{CIC} \cdot \left(\left\lceil \frac{N_{FIR} + 1}{2} \right\rceil + 1 \right)$	$\frac{544}{F_s} = 261.5\mu s$
t_{II}^{Delay}	II of the delay memories at F_s	$\max(\Delta)$	$\frac{1023}{F_s} = 491.18\mu s$
t_{II}^{Sum}	II of the cascaded sums	$2 \cdot \lceil \log_2(N_{am}) \rceil$	$\frac{12}{F_s} = 5.77\mu s$
t_s	Sensing time	$D_{CIC} \cdot D_{FIR} \cdot (N_s - 1)$	$\frac{4032}{F_s} = 1.94ms$
$t_{II}^{Filtering}$	II of the filter stage	$t_{II}^{CIC} + t_{II}^{DC} + t_{II}^{FIR}$	$\frac{587}{F_s} = 282.1\mu s$
$t_{II}^{Beamforming}$	II of the beamforming stage	$t_{II}^{Delay} + t_{II}^{Sum}$	$\frac{1035}{F_s} = 497.59\mu s$
t_{II}^{Power}	II of the SRP	2	$\frac{2}{F_s} = 0.96\mu s$
t_{II}	Sum of II's	$t_{II}^{Beamforming} + t_{II}^{Filtering} + t_{II}^{Power}$	$\frac{1199}{F_s} = 599.5\mu s$
t_{P-SRP}	Time to obtain a complete P-SRP	$N_o \cdot (t_{II} + t_s)$	$\frac{361984}{F_s} = 174.03ms$

TABLE 9: Performance analysis of the optimized designs when applying and combining the performance strategies. The values are expressed in ms.

	Initial	Continuous	Parallel continuous time multiplexing
t_{P-SRP}	174 ms	142.1 ms	23.7 ms

5.2.6. Performance Analysis. Table 10 summarizes the achievable performance based on the performance strategy. Notice that the power-efficient architecture analysed here achieves a higher performance than the high-performance architecture in [9, 10] when no strategy is applied. The performance difference is because the configuration of the power-efficient architecture under evaluation targets a different F_{max} . The difference between both F_{max} values leads to different F_s and D_F , which directly affects t_{P-SRP} .

The performance strategies are less effective for the power-efficient architecture. Although the initialization time is reduced when continuously acquiring data, it is still higher than that in the high-performance architecture described in [9, 10] because of the $t_{II}^{Filtering}$ overhead.

The last strategy proposed in Section 5.1 fully exploits the limited resources. The achievable performance is several orders of magnitude lower than the one of the high-performance architecture described in [9, 10]. This is mainly because of two factors. Firstly, the evaluation of the power-efficient architecture is done on a Flash-based

FPGA with a lower amount of resources than the one used to evaluate the one in [9, 10]. Secondly, the power-efficient architecture presents certain performance limitations which cannot be fully solved through the performance strategies.

The evaluation of the power-efficient architecture and the analysis done in [10], targeting different application requirements and with different types of FPGAs, demonstrate how the architectures and their performance strategies can be applied to any FPGA-based architecture.

5.3. Summary. The power-efficient architecture represents an alternative when power consumption is a priority. Due to its low demand in resources, this architecture is able to be implemented on Flash-based FPGAs. These FPGAs are very power efficient, but they lack in the number of available resources. Reduced performance of the sound locator is the price to pay for the power efficiency. Although several strategies can certainly accelerate the architecture's response, the limited FPGA resources and the architecture characteristics bound further acceleration.

6. Comparison of the Reconfigurable Architectures

This section presents a comparison of the high-performance architecture described in [10] and the proposed power-efficient architecture detailed in Section 5. Both architectures are evaluated on a Zynq-based platform

TABLE 10: Summary of the achievable performance and the related parameters when using the performance strategies proposed in Section 5.1.

Strategy	Parameter	Active subarrays			
		Inner 4 mics	Inner 12 mics	Inner 28 mics	All 52 mics
None	F_p (MHz)			F_s	
	Performance (Or/s)			$3.98E + 02$	
Continuous processing	F_p (MHz)			F_s	
	Performance (Or/s)			$4.50E + 02$	
Parallel continuous time multiplexing	F_{op} (MHz)	93.11	88.02	91.08	86.92
	$N_{BStages}$	6	6	6	5
	Performance (Or/s)	$2.70E + 03$	$2.70E + 03$	$2.70E + 03$	$2.25E + 03$

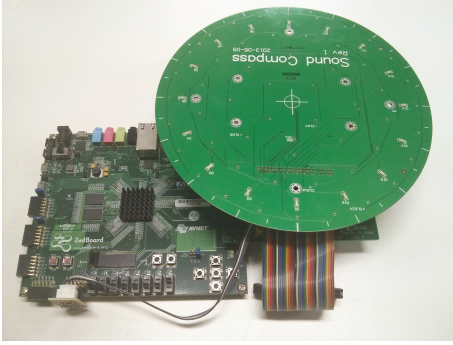


FIGURE 19: Demonstrator [41] and target platform on which both reconfigurable architectures are compared.

(Figure 19) for a fair comparison. Although both architectures have different goals, their characteristics make them both valid solutions for most of the acoustic applications, as far as they do not prioritize performance neither power efficiency. A final comparison against the state-of-the-art of related architectures is also briefly presented at the end of this section.

6.1. Frequency Response. The high-performance and power-efficient architectures use D_p to properly evaluate the quality of the array's response. Instead of evaluating one sound source at one particular orientation, like done in Section 5.2.1, the directivity is evaluated by placing a sound source at all the 64 supported orientations. The average of all directivities along with the 95% confidence interval is calculated for the supported orientations. Figure 20(a) depicts the resulting directivities based on the active subarrays for the proposed architecture. Notice that, when only the 4 inner microphones are enabled on both architectures, the predefined threshold of 8 for D_p is achieved by none of the architectures. The directivity increases in case 12 microphones are enabled, reaching the value of 8 at 3.1 kHz. This value is reached at 2.1 kHz and 1.7 kHz when 28 and all 52 microphones are enabled. One can also note that the 95% confidence interval noticeably increases at 4 kHz, 6 kHz, and 7 kHz for, respectively, the inner 4, 12, and 28 and all 52 microphones.

The power-efficient architecture outperforms the frequency response of the high-performance architecture, which is depicted in Figure 20(b). The selection of the

low-power FIR filter characteristics and the decomposition of D_F in the PDM demodulation have a lower impact on the power-efficient architecture. A high D_{CIC} leads to a higher-order low-pass FIR filter, improving the frequency response. The high-performance architecture leads to a lower resolution in the beamforming stage since the values of Δ are directly related to the input data rate. Therefore, besides the fact that a high D_{CIC} leads to a better frequency response, the variance of D_p of the architecture based on the sound source location increases with the sound source frequency, to become very sensitive to the beamed orientation. A possible solution is to implement a partially parallel low-pass FIR filter, reducing the existing dependency between D_{CIC} and N_{FIR} . It would increase the already high resource consumption of the filter stage.

The power-efficient architecture has higher beamforming resolution thanks to beamforming before downsampling the input data. Instead, the high-performance architecture performs the beamforming after the filter stage, whose data has a lower rate. The capacity of properly determining the DoA increases with N_{am} for both architectures, as shown in Figure 20.

6.2. Resource Consumption. The power-efficient architecture presents a significantly lower resource consumption compared to the high-performance architecture. Figure 21 depicts a comparison of the resource consumption of both architectures when targeting a Zynq 7020 FPGA. Although the low resource consumption of the power-efficient architecture allows the use of a smaller and lower demanding power FPGA, the Zynq 7020 FPGA is used in order to fairly compare both architectures. The amount of different types of resources demanded by the proposed architecture is significantly lower than the architecture presented in [9, 10]. The low resource consumption is possible thanks to the reduction of the number of filter chains, leading to a more efficient beamforming operation in terms of resources. Whereas each microphone in the high-performance architecture has an individual filter chain, the power-efficient architecture only needs one.

This percentage decreases to 14.7% and 32.8% of the consumed registers and Look-Up Tables (LUTs), respectively, in the power-efficient architecture. An efficient

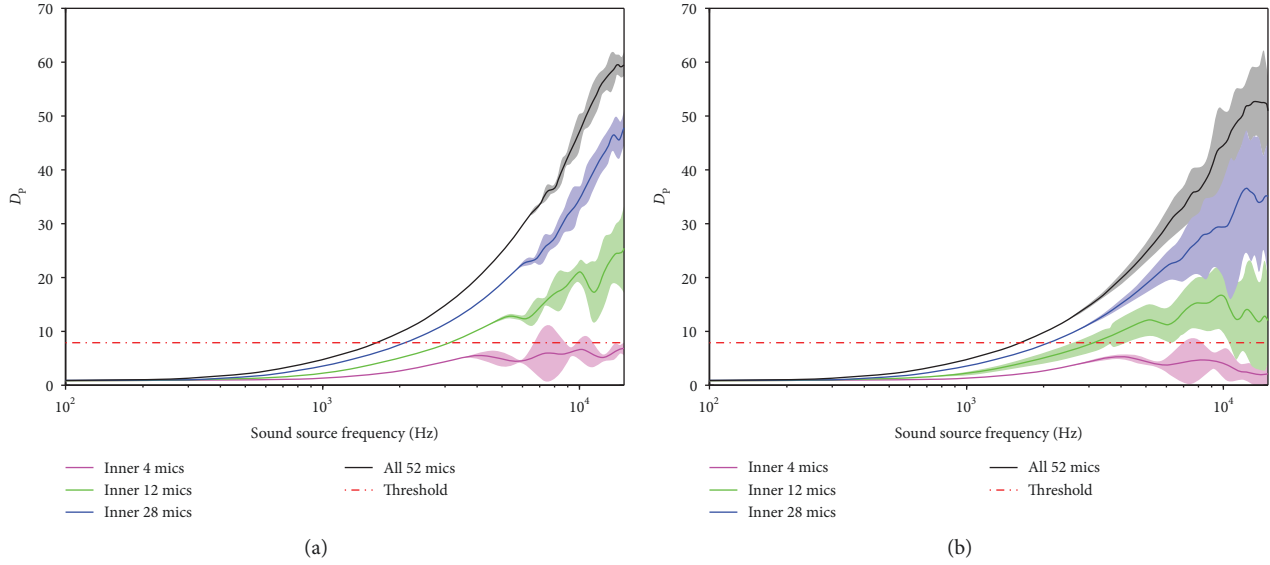


FIGURE 20: Average D_p with a 95% confidence interval for the supported orientations when combining subarrays of the power-efficient architecture (a) and the high-performance architecture (b).

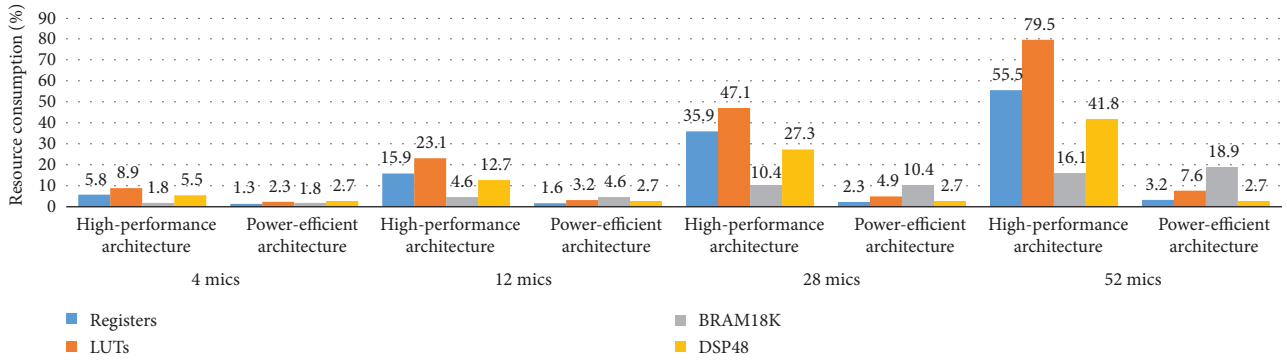


FIGURE 21: Comparison of both architectures based on the Zynq 7020 resource consumption after placement and routing when combining microphone subarrays.

memory partition is possible thanks to the storage of PDM signals and the use of LUTs as internal memory. LUTs are the constraint resource in the high-performance architecture, justifying the increment of the consumption of dedicated internal memories (BRAMs). In this regard, BRAMs are the limiting resource for the power-efficient architecture. Due to the relatively low consumption of LUTs, the consumption of BRAMs can be certainly reduced if LUTs are used as internal memory in the beamforming stage. In fact, BRAMs are only required to store the incoming samples from the 24 microphones of the outer subarray. The available resources in the Zynq 7020 support up to 11 instantiations of the power-efficient architecture, which represents the capacity to compute the incoming signal from more than 500 microphones simultaneously.

6.3. Timing and Performance. Figure 22 depicts a comparison of t_{P-SRP} for both architectures defined by the supported configurations detailed in Section 3.2.2. No performance strategy has been applied. Notice how the values of t_{P-SRP} differ

with approximately 33 ms for each supported configuration. Although t_{P-SRP} is equally defined for both architectures, the II values of the beamforming stage vary due to $\max(\Delta)$, which is also reflected in the memory requirements (Figure 9).

The placement of the beamforming stage into the architecture does not only affect the frequency response but also directly determines the achievable performance. The performance strategies, proposed in [10] and in Section 5.1, accentuate the architectural differences. Figure 14 exemplifies the relevance of the beamformer placement from the performance perspective. Although both architectures support the same performance strategy of the continuous processing, the position of the beamforming stage determines what II of the components affects the increment of t_{P-SRP} . Whereas in the high-performance architecture only the II of the power stage increases t_{P-SRP} , the II of the filter stage and additional clock cycles to reset the filters also increase t_{P-SRP} in the power-efficient architecture.

Figure 23 shows the values of t_{P-SRP} for the same design space when applying the parallel continuous time

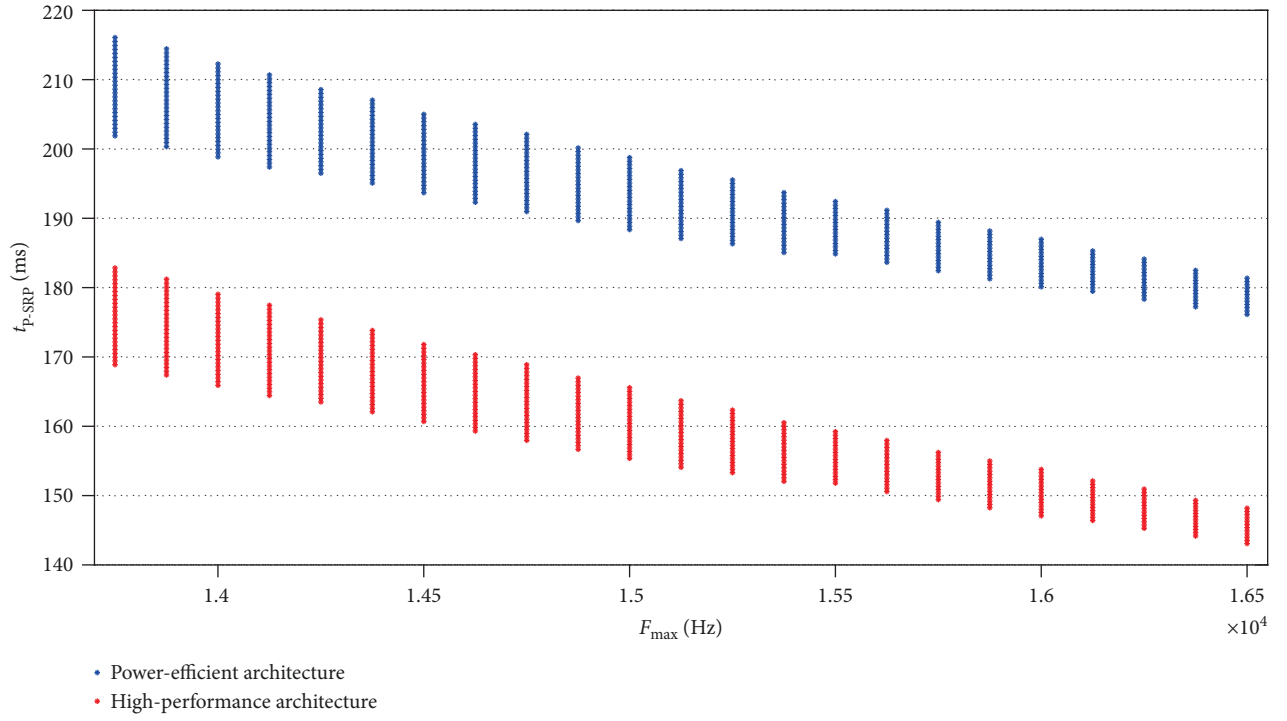


FIGURE 22: Evolution of t_{P-SRP} for each architecture when no performance strategy is applied. The explored design space corresponds to the one depicted in Figure 6.

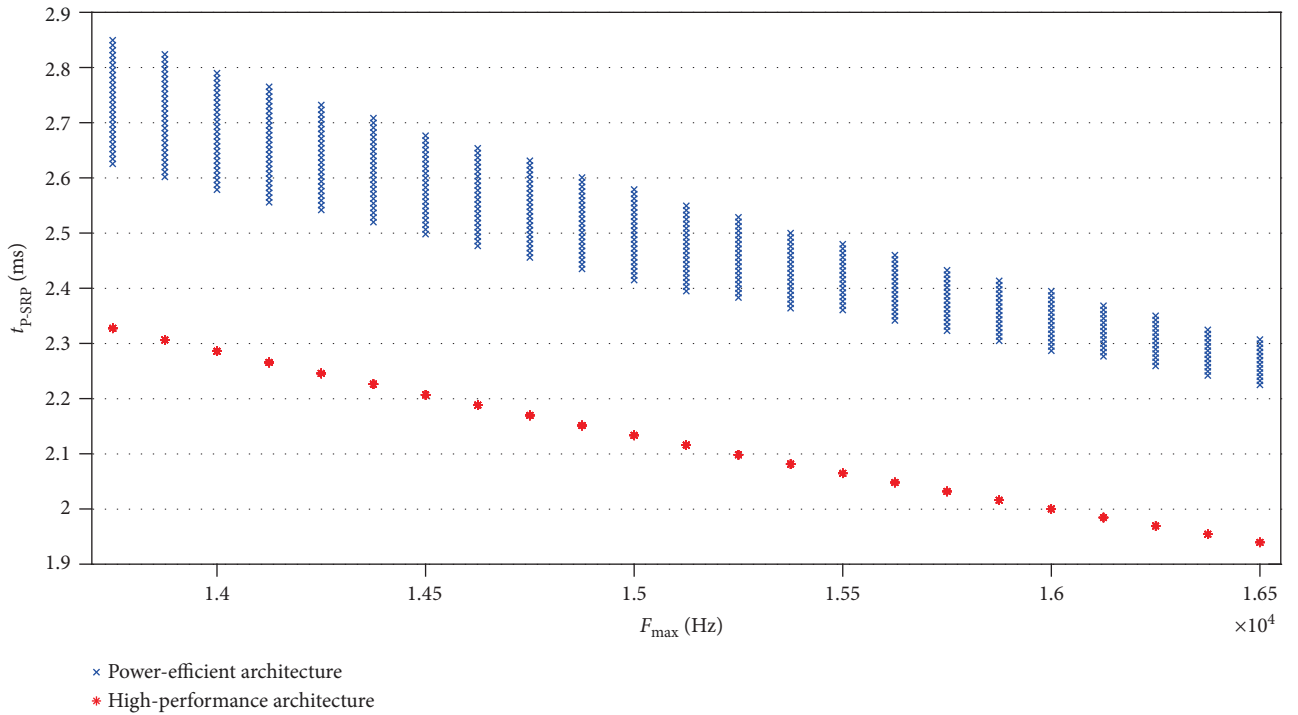


FIGURE 23: Evolution of t_{P-SRP} for each architecture when the best performance strategy is applied. The explored design space corresponds to the one depicted in Figure 6.

multiplexing performance strategy. The variation of t_{P-SRP} based on F_{max} reflects the dependency of t_{P-SRP} on F_s for the power-efficient architecture. This dependency, however, disappears for the high-performance architecture,

where t_{P-SRP} only depends on F_{max} . Such a characteristic reduces the dependency on F_s , which corresponds to the microphone's clock, during the design stage of a high-performance architecture.

TABLE 11: Comparison of architectures when applying the parallel continuous time multiplex strategy. The maximum number of beamformers is obtained based on the available resources and the resource consumption of each beamformer (Table 6 and reported in [10]). The value of F_{op} corresponds to the maximum frequency reported by the Vivado 2016.4 tool after placement and routing.

	High-performance architecture [10]				Power-efficient architecture			
	4 mics	12 mics	28 mics	52 mics	4 mics	12 mics	28 mics	52 mics
$N_{BStages}$	55	23	10	6	170	117	59	11
$N_{FStages}$	—	—	—	—	70	70	70	70
F_{op} (MHz)	95.62	93.27	91.97	87.91	165.04	141.70	112.91	94.22
Performance (Or/s)	$8.35E+07$	$3.41E+07$	$1.46E+07$	$8.37E+06$	$5.36E+06$	$3.69E+06$	$1.86E+06$	$3.47E+05$

TABLE 12: Comparison of the time performance and the reported power consumption of the microphone array and the FPGA.

References	Year	Device	Microphone	N_{Mics}	Time (ms)	Time/mic (ms/mic)	Power (mW)	Power/mic (mW/mic)
[20]	2015	IGLOO2	STMicroelectronics MP34DT01	8	—	—	30.44	3.80
[21]	2016	Spartan-6	STMicroelectronics MP32DB01	16	18.85	1.18	78.99	4.94
[42]	2016	EFM32	InvenSense INMP504	4	249	62.25	7.20	1.80
[10]	2017	Zynq 7020	Analog Devices ADMP521	52	2	0.04	343.92	6.61
[8]	2018	SmartFusion2 M2S050	InvenSense ICS-41350	52	141.66	2.724	33.78	0.65
Power-efficient architecture	2019	SmartFusion2 M2S025	InvenSense ICS-41350	52	23.68	0.45	63.98	1.23

Table 11 summarizes the parameters involved on the calculation of this performance strategy for both architectures. The achievable performance expressed in *Or/s* decreases when increasing N_{am} on both architectures. Notice that, although the power-efficient architecture is able to operate at a higher frequency and to allocate more instances, the achievable performance of the high-performance architecture outperforms the power-efficient architecture despite both target the same FPGA.

6.4. Summary. Table 12 summarizes the comparison of the proposed power-efficient architecture and the related works from a timing and power consumption point of view. As a consequence of the lower resource consumption, not only larger microphone arrays can be processed in parallel but also more power-efficient FPGAs can be used to minimize the power consumption. The main difference in power consumption between the analysed architecture and the one described in [8] is in the microphones' power consumption due to operating at a different operational mode. The power-efficient architecture presents a major reduction of the power consumption when compared to the high-performance architecture described in [10], achieving the lowest power per microphone ratio when all the subarrays are active. Although the power-efficient architecture is slower than the high-performance architecture, the time-per-microphone ratio is better than other related solutions.

7. Conclusions

FPGA-based embedded systems offer sufficient flexibility to support dynamic acoustic beamformers able to achieve real-time performance or power efficiency. Nevertheless, these desirable features are only achievable through design considerations and performance strategies to fully exploit the FPGA's characteristics. On the one hand, design considerations lead to compromises on the selection of the architecture's components, grouped in stages, to obtain the desired response. The position of the beamforming stage, for instance, does affect not only memory requirements but also performance, frequency response, and resource consumption of the architecture. The specifications of the PDM demodulation present a limited impact on the performance and resource consumption. On the other hand, performance strategies enable a higher performance by exploiting the FPGA's resources. Although such performance strategies are strongly dependent on the reconfigurable architecture, they are capable to significantly increase the performance of reconfigurable acoustic beamforming architectures.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the European Regional Development Fund (ERDF) and the Brussels-Capital Region-Innoviris within the framework of the Operational Programme 2014-2020 through the ERDF-2020 project ICITY-RDI.BRU.

References

- [1] B. Widrow and F. L. Luo, "Microphone arrays for hearing aids: an overview," *Speech Communication*, vol. 39, no. 1-2, pp. 139–146, 2003.
- [2] A. Izquierdo-Fuente, L. Del Val, M. I. Jiménez, and J. J. Villacorta, "Performance evaluation of a biometric system based on acoustic images," *Sensors*, vol. 11, no. 10, pp. 9499–9519, 2011.
- [3] L. del Val, A. Izquierdo-Fuente, J. J. Villacorta, and M. Raboso, "Acoustic biometric system based on preprocessing techniques and linear support vector machines," *Sensors*, vol. 15, no. 6, pp. 14241–14260, 2015.
- [4] M. Akay, A. Dragomir, Y. M. Akay et al., "The assessment of stent effectiveness using a wearable beamforming MEMS microphone array system," *IEEE Journal of Translational Engineering in Health and Medicine*, vol. 4, pp. 1–10, 2016.
- [5] D. C. Moore and I. A. McCowan, "Microphone array speech recognition: experiments on overlapping speech in meetings," in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03)*, pp. V-497, Hong Kong, China, April 2003.
- [6] S. Doclo and M. Moonen, "GSVD-based optimal filtering for single and multimicrophone speech enhancement," *IEEE Transactions on Signal Processing*, vol. 50, no. 9, pp. 2230–2244, 2002.
- [7] M. Brandstein and D. Ward, *Microphone Arrays: Signal Processing Techniques and Applications*, Springer Science & Business Media, 2013.
- [8] B. da Silva, L. Segers, A. Braeken, K. Steenhaut, and A. Touhafi, "A Low-power FPGA-based architecture for microphone arrays in wireless sensor networks," in *Applied Reconfigurable Computing. Architectures, Tools, and Applications: 14th International Symposium. ARC 2018* pp. 281–293, Springer, Cham.
- [9] B. da Silva, L. Segers, A. Braeken, and A. Touhafi, "Runtime reconfigurable beamforming architecture for real-time sound-source localization," in *2016 26th International Conference on Field Programmable Logic and Applications (FPL)*, pp. 1–4, Lausanne, Switzerland, August–September 2016.
- [10] B. da Silva, A. Braeken, K. Steenhaut, and A. Touhafi, "Design considerations when accelerating an FPGA-based digital microphone array for sound-source localization," *Journal of Sensors*, vol. 2017, Article ID 6782176, 20 pages, 2017.
- [11] J. Tiete, F. Domínguez, B. da Silva, A. Touhafi, and K. Steenhaut, "MEMS microphones for wireless applications," in *Wireless MEMS Networks and Applications*, pp. 177–195, Elsevier, 2017.
- [12] Microsemi, "Alexa voice service development kit (ZLK38AVS2)," July 2018, <https://www.microsemi.com/productdirectory/connected-home/4628-zlk38avs>.
- [13] B. da Silva, A. Braeken, and A. Touhafi, "FPGA-based architectures for acoustic beamforming with microphone arrays: trends, challenges and research opportunities," *Computers*, vol. 7, no. 3, p. 41, 2018.
- [14] E. Zwyssig, M. Lincoln, and S. Renals, "A digital microphone array for distant speech recognition," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 5106–5109, Dallas, TX, USA, March 2010.
- [15] I. Hafizovic, C. I. C. Nilsen, M. Kjølnerbakken, and V. Jahr, "Design and implementation of a MEMS microphone array system for real-time speech acquisition," *Applied Acoustics*, vol. 73, no. 2, pp. 132–143, 2012.
- [16] A. Izquierdo, J. Villacorta, L. del Val Puente, and L. Suárez, "Design and evaluation of a scalable and reconfigurable multi-platform system for acoustic imaging," *Sensors*, vol. 16, no. 10, p. 1671, 2016.
- [17] I. Salom, V. Celebic, M. Milanovic, D. Todorovic, and J. Prezelj, "An implementation of beamforming algorithm on FPGA platform with digital microphone array," in *Audio Engineering Society Convention 138*, Audio Engineering Society, 2015.
- [18] D. Todorović, I. Salom, V. Celebić, and J. Prezelj, "Implementation and application of FPGA platform with digital MEMS microphone array," in *Proceedings of the Proceedings of 4th International Conference on Electrical, Electronics and Computing Engineering*, pp. 5–8, Kladovo, Serbia, 2017.
- [19] N. Hegde, *Seamlessly interfacing MEMS microphones with Blackfin processors*, EE-350 Engineer-to-Engineer Note, 2010.
- [20] L. Petrica and G. Stefan, "Energy-efficient WSN architecture for illegal deforestation detection," *International Journal of Sensors and Sensor Networks*, vol. 3, no. 3, pp. 24–30, 2015.
- [21] L. Petrica, "An evaluation of low-power microphone array sound source localization for deforestation detection," *Applied Acoustics*, vol. 113, pp. 162–169, 2016.
- [22] J. Tiete, F. Domínguez, B. da Silva, L. Segers, K. Steenhaut, and A. Touhafi, "SoundCompass: a distributed MEMS microphone array-based sensor for sound source localization," *Sensors*, vol. 14, no. 2, pp. 1918–1949, 2014.
- [23] T. E. Bogale, L. Vandendorpe, and L. B. Le, "Sensing throughput tradeoff for cognitive radio networks with noise variance uncertainty," in *Proceedings of the 9th International Conference on Cognitive Radio Oriented Wireless Networks*, pp. 435–441, Oulu, Finland, 2014.
- [24] T. Taguchi, T. Nakadai, R. Egusa et al., "Investigation on optimal microphone arrangement of spherical microphone array to achieve shape beamforming," in *2014 5th International Conference on Intelligent Systems, Modelling and Simulation*, pp. 330–333, Langkawi, Malaysia, January 2014.
- [25] A. Malgoezar, M. Snellen, P. Sijtsma, and D. Simons, "Improving beamforming by optimization of acoustic array microphone positions," in *Proceedings of the 6th Berlin Beamforming Conference*, pp. 1–24, Berlin, Germany, 2016.
- [26] E. Sarradj, "A generic approach to synthesize optimal array microphone arrangements," in *6th Berlin beamforming conference, BeBeC-2016-S4*, p. 12, Berlin, Germany, 2016.
- [27] Analog Devices, "ADMP521 datasheet," July 2018, <http://www.analog.com/media/en/technical-documentation/obsolete-data-sheets/ADMP521.pdf>.

- [28] TDK Invensense, “INMP621 datasheet,” July 2018, <https://www.invensense.com/wp-content/uploads/2015/02/INMP621.pdf>.
- [29] Knowles, “SPH0645LM4H-B datasheet,” July 2018, <https://www.alldatasheet.com/datasheet-pdf/pdf/791053/KNOWLES/SPH0645LM4H-B.html>.
- [30] TDK Invensense, “ICS-43434 datasheet,” July 2018, <https://www.invensense.com/wp-content/uploads/2016/02/DS000069-ICS-43434-v1.2.pdf>.
- [31] E. Hogenauer, “An economical class of digital filters for decimation and interpolation,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 2, pp. 155–162, 1981.
- [32] M. P. Donadio, “CIC filter introduction,” in *IEEE International Symposium on Communications*, pp. 1–6, 2000.
- [33] R. Lyons, “Understanding cascaded integrator-comb filters,” *Embedded Systems programming*, vol. 18, pp. 14–27, 2005.
- [34] M. J. Taghizadeh, P. N. Garner, and H. Bourlard, “Microphone array beampattern characterization for hands-free speech applications,” in *2012 IEEE 7th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, pp. 465–468, Hoboken, NJ, USA, June 2012.
- [35] H. L. Van Trees, *Optimum Array Processing: Part IV of Detection, Estimation, and Modulation Theory*, John Wiley & Sons, 2004.
- [36] Microsemi SmartFusion2 SoC FPGA and IGLOO2 FPGA Fabric, “User Guide 0445 (UG0445) v.6,” July 2018, http://www.microsemi.com/index.php?option=com_docman&task=doc_download&gid=132008.
- [37] Microsemi SmartFusion2 SoC and IGLOO2 FPGA Low-Power Design, “User Guide 0444 (UG0444) v.5,” July 2018, http://www.microsemi.com/document-portal/doc_download/132010-ug0444-smartfusion2-andigloo2-fpga-low-power-design-user-guide.
- [38] TDK Invensense, “ICS41350 datasheet,” July 2018, <https://www.invensense.com/wp-content/uploads/2016/02/DS000047-ICS-41350-v1.1.pdf>.
- [39] B. da Silva, L. Segers, Y. Rasschaert, Q. Quevy, A. Braeken, and A. Touhafi, “A Multimode SoC FPGA-based acoustic camera for wireless sensor networks,” in *2018 13th International Symposium on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC)*, pp. 1–8, Lille, France, July 2018.
- [40] J. Lewis, “Analog and digital MEMS microphone design considerations,” July 2018, <http://www.analog.com/media/en/technical-documentation/technical-articles/Analog-and-Digital-MEMS-Microphone-DesignConsiderations-MS-2472.pdf>.
- [41] B. da Silva, L. Segers, A. Braeken, and A. Touhafi, “A runtime reconfigurable FPGA-based microphone array for sound source localization,” in *2016 26th International Conference on Field Programmable Logic and Applications (FPL)*, p. 1, Lausanne, Switzerland, August-September 2016.
- [42] G. Ottoy, B. Thoen, and L. De Strycker, “A low-power MEMS microphone array for wireless acoustic sensors,” in *2016 IEEE Sensors Applications Symposium (SAS)*, pp. 1–6, Catania, Italy, April 2016.

