Hindawi
Journal of Sensors
Volume 2017, Article ID 6782176, 20 pages
https://doi.org/10.1155/2017/6782176

Research Article

Design Considerations When Accelerating an FPGA-Based Digital Microphone Array for Sound-Source Localization

Bruno da Silva, An Braeken, Kris Steenhaut, and Abdellah Touhafi

INDI Department, Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussels, Belgium

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

Received 16 March 2017; Accepted 16 May 2017; Published 20 June 2017

Academic Editor: Paolo Bruschi

The use of microphone arrays for sound-source localization is a well-researched topic. The response of such sensor arrays is dependent on the quantity of microphones operating on the array. A higher number of microphones, however, increase the computational demand, making real-time response challenging. In this paper, we present a Filter-and-Sum based architecture and several acceleration techniques to provide accurate sound-source localization in real-time. Experiments demonstrate how an accurate sound-source localization is obtained in a couple of milliseconds, independently of the number of microphones. Finally, we also propose different strategies to further accelerate the sound-source localization while offering increased angular resolution.

1. Introduction

Most of the signal processing needed in microphone arrays is traditionally done using general purpose processors. However, the computational demand is directly related to the number of microphones of the array. This number is drastically increasing as low-cost MEMS technology is readily available. Current FPGAs are a potential solution thanks to their high-computational power and low latency response. In fact, FPGAs have been already considered by other researchers, mainly for converting the analogue or digital microphone signals into an audio format [1, 2] without further signal processing computation. We believe that FPGAs not only are able to manage relatively large microphone arrays, but also enable a faster response when compared to using general purpose processors.

In order to satisfy the most time stringent sound-source localization applications that also use an incremental number of microphones, we propose a flexible, scalable, and real-time architecture. Main targets are the performance, scalability, and accuracy of the system to detect the direction of sound sources in real-time. Furthermore, we propose several techniques based on our architecture to accelerate the sound-source localization to guarantee real-time detection.

The architecture presented in this paper is an improved and more detailed version than the one presented in [3]. Because this novel architecture is designed to be part of an embedded system, the resource and the power consumption are included together with the performance in our analysis of the system. A frequency analysis is also done based on design parameters such as the number of microphones or the number of orientations. Altogether this leads to an architecture for which the frequency response must satisfy the basic needs of an application requiring real-time sound-source localization.

The main contributions of this work can be summarized as follows:

(i) A Filter-and-Sum based architecture for a fast sound-source localization.

(ii) A complete frequency and performance analysis of the system.

(iii) Strategies to speed up the overall execution time.

This paper is organized as follows. Section 2 presents related work. The principles used for the sound-source localization are introduced in Section 3. In Section 4 our proposed architecture is detailed. A complete time analysis

and different strategies to increase performance are presented in Section 5. In Section 6 the proposed architecture is analysed. Finally, the conclusions are drawn in Section 7.

2. Related Work

The use of microphone arrays for sound-source localization is a well-researched problem, where complexity increases with the number of microphones involved and the required response time of the application. The response time is indeed crucial for applications such as a counter-sniper systems [4, 5]. Such military systems are composed of microphone arrays mounted on top of a soldiers helmet and connected to an FPGA for signal processing. A similar approach is applied in [6], where the authors present a hat-type hearing system composed of 48 digital MEMs microphone array with an FPGA as the computational component. Their main target is a hearing aid system which emphasizes up to 10 dB the sound coming from a certain direction. Such type of applications demands a fast response of the system while being power efficient.

Indoor applications, such as videoconferencing, home surveillance, and patient care, make also use of microphone arrays for speech detection [1, 7]. This paper describes the design and implementation on an FPGA of an eight-element digital MEMS microphone array for distant speech recognition. In [8] the authors propose a beamforming-based acoustic system for localization of the dominant noise source. The signal acquisition consists of a microphone array composed of up to 33 MEMS microphones whereas the PDM demodulation and the beamforming are implemented in an FPGA. The implementation in the FPGA is completed with the delay-and-sum beamforming, measuring 60 angles, and generating a polar map for directivity pattern presentation. Another example is proposed in [9], in which the sound-source localization is obtained by using distributed microphone arrays in a WSN. The distributed information collected by the nodes is transferred and processed using data-fusion techniques in order to locate and profile the sound sources. Despite the fact that they implement most of the processing components on an FPGA, the 64k-FFT component becomes too large and resource hungry such that it is not suitable for low and middle-end FPGAs. In both publications, however, their solutions are not scalable and not adaptable to dynamic acoustic environments. Furthermore, they do not provide information about how fast their systems can be. Instead, we present a detailed description and analysis of a flexible, scalable, and real-time architecture.

3. Sound-Source Localization

Our microphone array is designed to spatially sample its surrounding sound field in order to detect and to locate certain types of sound sources. A 360° sound power scan is performed for a configurable number of orientations. A beamforming technique focuses the array in one specific direction or orientation, by amplifying all sounds coming from that direction and by suppressing sounds coming from other directions. A *polar power plot* is obtained from which the lobes can be used to estimate the nearby sound sources. Figure 1 shows the functional elements required to locate the sound-source, which involve several filters, a beamformer, and a relative sound power estimator.

3.1. Microphone Array Description. The sensor array is composed of 52 digital MEMS microphones and designed for far-field and nondiffuse sound fields [9]. The array pattern consists of four concentric subarrays of 4, 8, 16, and 24 MEMS microphones mounted on a 20 cm circular printed board (Figure 2). Each subarray is differently positioned in order to facilitate the capture of spatial acoustic information using a beamforming technique. Furthermore, the sensor array response is dynamically modified by individually activating or deactivating subarrays. This distributed geometry allows adapting the sensor to different sound sources. For instance, not all the subarrays need to be active to detect a particular sound-source. The computational requirements drastically decrease and the sensor array becomes more power efficient if only a few numbers of subarrays are active.

3.2. Filters. The selected digital MEMS microphones are the ADMP521 MEMS microphones designed by Analog Devices, which offer an omnidirectional polar response and a wide-band frequency response ranging from 100 Hz up to 16 kHz [10]. These digital MEMS microphones have a multiplexed pulse density modulation (PDM) as output. The PDM signals are generated by using an analogue to digital converter (ADC) based on a sigma delta converter. The sigma delta conversion technique uses an embedded integrator-comparator circuit to sample the analogue signal and outputs a 1-bit signal [11]. The ADMP521 MEMS microphones use a fourth-order sigma delta converter, which reduces the added noise in the audio frequency spectrum by shifting it to higher frequency ranges. This undesirable high-frequency noise needs to be removed. The ADMP521 MEMS microphones require a clock input of around 1 to 3 MHz as sampling frequency (F_S). 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. Therefore, the PDM signal needs not only to be filtered to remove the noise but also to be downsampled to convert the audio signal to a Pulse-Code Modulation (PCM) format. The target audible frequency range, from $F_{\min}$ to $F_{\max}$, determines the decimation factor (D_F) to properly downsample the PDM signal while satisfying the Nyquist theorem.

$$D_F = \left\lceil \frac{F_S}{2 \cdot F_{\max}} \right\rceil. \tag{1}$$

The usual range of D_F is from a few tens up to hundreds when targeting audible frequency ranges. For instance, D_F needs to be 83 to recover audio signal oversampled at 2.49 MHz for a target $F_{\max}$ of 15 kHz.

3.3. Filter-and-Sum Beamforming. The beamforming technique applied in our proposed architecture is based on the Filter-and-Sum beamforming [12]. The original Filter-and-Sum beamforming applies an independent weight to

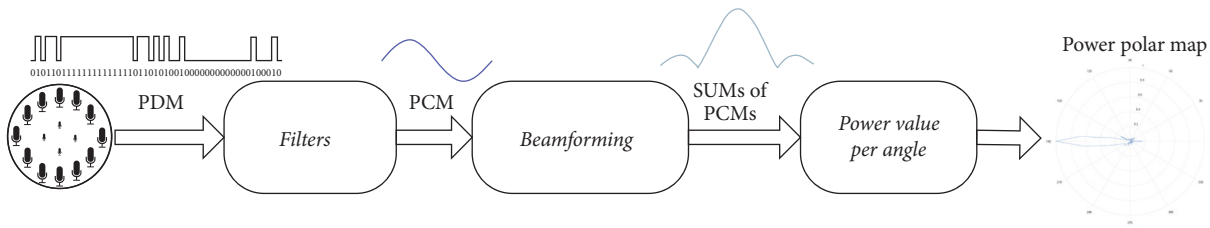

Figure 1: Operations needed for the proposed architecture to locate a sound-source.

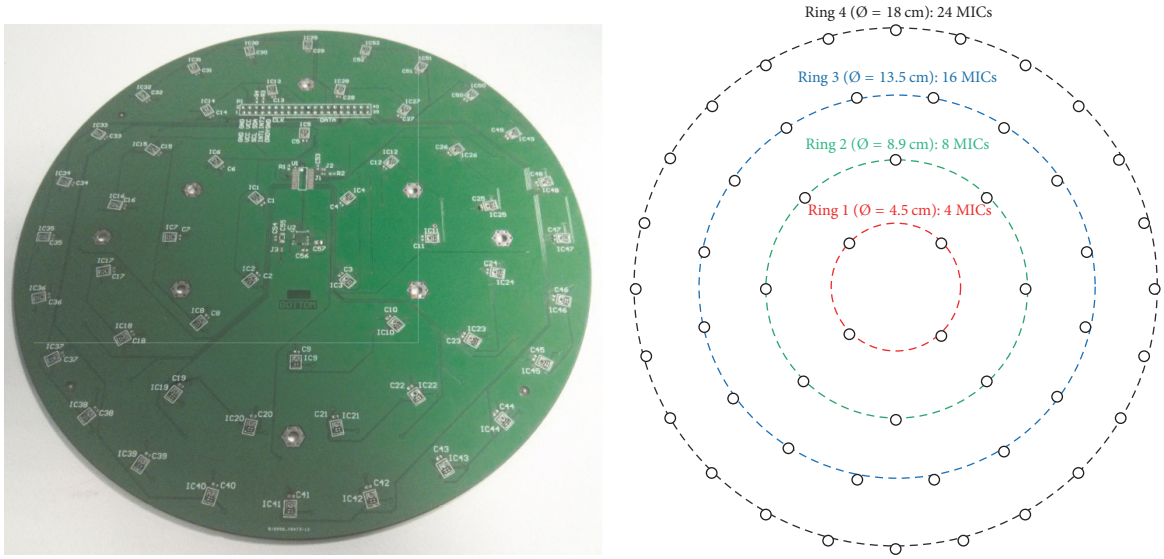

Figure 2: Sound-source localization device composed of 4 MEMS microphone subarrays.

each microphone output before summing them. The overall effect is an amplification of the signal coming from a target orientation while suppressing signals from other orientations. A variant version of the Filter-and-Sum recovers the audio signal from the PDM signal, applies the same low-pass FIR filter, and delays the filter output signal of each microphone by a specific amount of time (Δ) before adding all the output signals together (Figure 3). The time delay (Δ_m) for a microphone m is determined by the focus direction θ, the position vector ($\overrightarrow{r_m}$) of microphone m, and the speed of sound (c).

$$\Delta_m = \frac{\overrightarrow{r_m} \cdot \vec{k}}{c}, \tag{2}$$

where the unitary vector ($\vec{k}$) defines the direction vector of a far-field propagating signal with a focus direction θ. The total output ($O(\theta, t)$) of the array can be expressed based on the signal output of each microphone in the time domain $s_m(t)$ and the number of microphones in the array (M):

$$O(\theta, t) = \sum_{m=1}^{M} s_m\left(t - \Delta_m(\theta)\right). \tag{3}$$

The response of the Filter-and-Sum beamforming, however, is usually represented in the frequency domain due to its dependence on the signal frequency. Let $S_m(\omega)$ be the output signal of each microphone at angular speed $\omega = 2\pi f$ for frequency f and M the number of microphones in the array. The total output ($O(\theta, \omega)$) is defined as in [13]:

$$O(\theta, \omega) = \sum_{m=1}^{M} S_m(\omega)\, e^{-j\omega\Delta_m(\theta)}. \tag{4}$$

which can be simplified by assuming a monochromatic acoustic wave as

$$\begin{aligned} O(\theta, \omega) &= S_o(\omega) \sum_{m=1}^{M} e^{jr_m\omega_n(\theta_0 - \theta)} \\ &= S_o(\omega)\, W(w_n, \theta_0, \theta), \end{aligned} \tag{5}$$

where $S_o(\omega)$ is the output signal of the monochromatic wave, w_n is the incoming monochromatic angular speed, θ_0 is its direction, and θ is the array focus. $W(w_n, \theta_0, \theta)$ is known as the array pattern, which determines the amplification or gain of the array output. For instance, when $\theta_0 = \theta$, which occurs when the array is focusing in the direction of the incoming monochromatic wave, the gain reaches its maximum M, equal to the number of microphones.

3.4. Polar Steered Response Power. The direction of the sound-source is located by measuring the relative sound

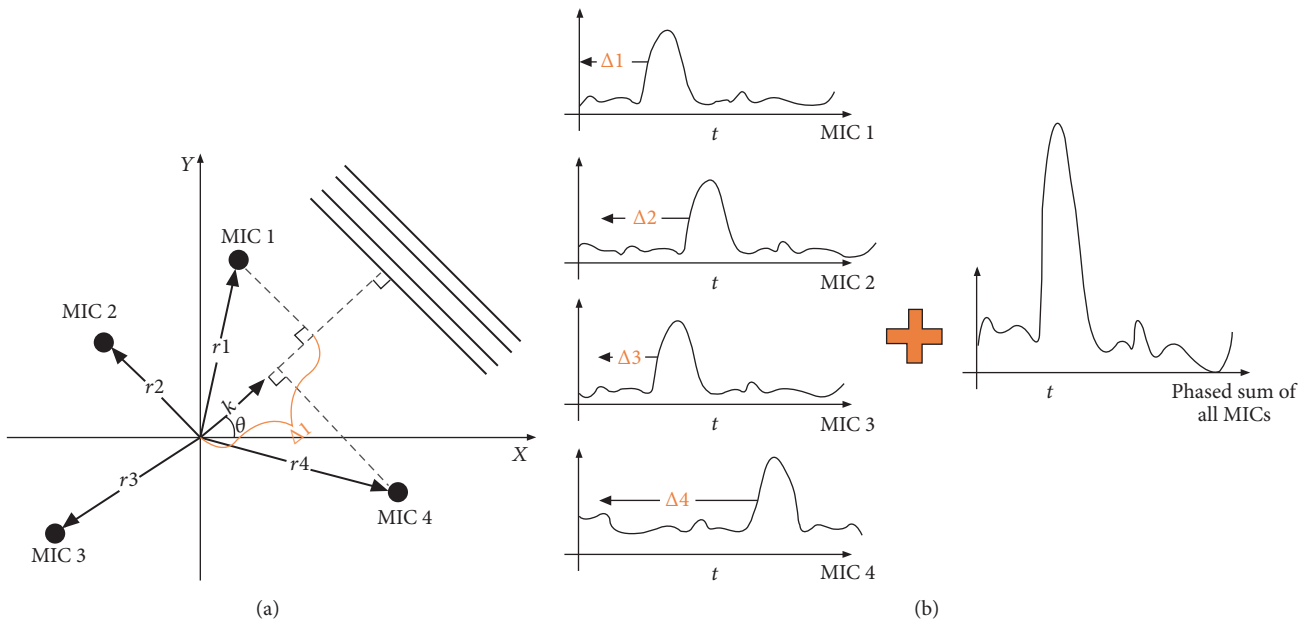

Figure 3: The proposed Filter-and-Sum beamforming filters and delays the output of each microphone before adding them together. (a) The acoustic wave received at each microphone is measured and filtered. The beamforming technique considers the time Δ_m that the input signal takes to travel from the microphone m to the origin is proportional to the projection of the microphone vector $\vec{r_m}$ on $\vec{k}$. (b) This Δ_m is determined by the position of the microphone in the array and the desired focus direction θ of the array. Consequently, the signals coming from the same direction are amplified after the addition of the delayed inputs. Source [9].

power per horizontal direction, which is done by a 360° sweep overview of the surrounding sound field. The directional power output of a microphone array, defined here as the *polar steering response power* (P-SRP), corresponds to the array's directional response to sound sources present in a sound field (Figure 4). The P-SRP is obtained by considering multiple broadband sources coming from different directions, for instance, human speech.

The output power when the microphone array is exposed to a broadband sound-source $S(w)$ with an angle of incidence θ_0 can be modelled as

$$O(\theta, S) = A_1 W(wn_1, \theta_0, \theta) + A_2 W(wn_2, \theta_0, \theta) + \cdots + A_n W(wn_n, \theta_0, \theta), \quad (6)$$

where A_i with $i \in \{1, \ldots, n\}$ is the amplitude of one of the n frequency components of $S(w)$. The equation can be generalized to consider a sound field ϕ composed of multiple broadband sound sources at different locations and with uncorrelated noise:

$$O(\theta, \phi) = O(\theta, S_1) + O(\theta, S_2) + \cdots + O(\theta, Sn) + \text{Noise}_{\text{uncorrelated}}. \quad (7)$$

The array's power output can be expressed as

$$P(\theta, \phi) = |O(\theta, \phi)|^2 \quad (8)$$

since the power of a signal is the square of the array's power output. Finally, the normalized power output is defined as the P-SRP.

$$\text{P-SRP}(\theta, \phi) = \frac{P(\theta, \phi)}{\max_{\theta \in [0, 2\pi]} P(\theta, \phi)}. \quad (9)$$

The comparison of $P(\theta, \phi)$ for different values of θ determines in which direction the sound-source is located since the maximum power is obtained when the focus corresponds to the location of a sound-source.

The calculation of the P-SRP is usually defined in the frequency domain [14, 15], which requires the computation of a Fourier transform. Instead, we propose applying Parseval's theorem which states that the sum of the squares of a function is equal to the sum of the squares of its transform. This theorem drastically simplifies the calculations since P-SRP can be computed in the time domain. Let us define the sensing time (t_s) as the time the array is registering the previously defined sound field ϕ for each orientation. Therefore, the power $P(\theta, t_s)$ can be expressed as follows:

$$P(\theta, t_s) = \frac{1}{t_s} \sum_{t=1}^{t_s} \left| O(\theta, t_\phi) \right|^2. \quad (10)$$

Consequently, P-SRP can be expressed in the time domain by

$$\text{P-SRP}(\theta, t_s) = \frac{P(\theta, t_s)}{\max_{\theta \in [0, 2\pi]} P(\theta, t_s)}. \quad (11)$$

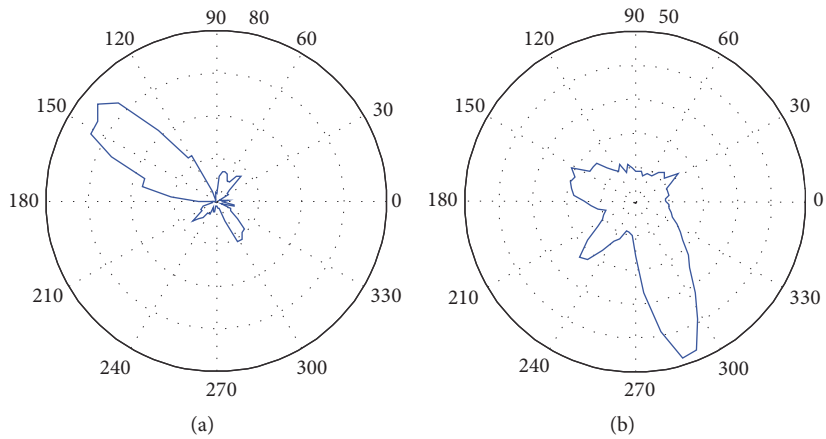

Figure 4: Examples of a polar map obtained under experimental conditions for sound sources of 5 kHz (a) and 8 kHz (b).

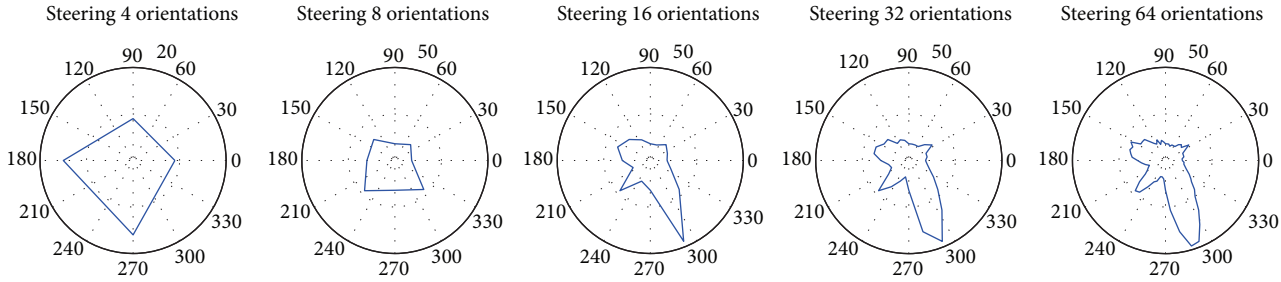

Figure 5: Examples of polar maps with different angular resolution locating a sound-source of 8 kHz. A low number of orientations clearly lead to wrong sound-source location.

3.5. Sensor Array Evaluation. The defined P-SRP allows estimating the direction of arrival of multiple sound sources under different sound field conditions. Nevertheless, the precision and accuracy of its estimation can be determined by different quality metrics.

The Filter-and-Sum beamforming is applied to a discrete number of orientations or angles. The angular resolution of the microphone array is determined by the number of measurements per 360° sweep. A higher number of measurements increment the resolution of the P-SRP displayed as a polar power map (Figure 5) and decrease the location error of the sound-source. The lobes of this polar power map can then be used to estimate the bearing of nearby sound sources in nondiffuse sound fields conditions. In fact, the characteristics of the main lobe when considering a single sound-source scenario determine the directivity of the microphone array. The definition of array directivity, D_p, is proposed in [16] for broadband signals. The authors propose the use of (D_p) as a metric of the quality of the array since D_p depends on the main lobe shape and its capacity to unambiguously point to a specific bearing. The definition of array directivity presented in [16] is adapted for 2D polar coordinates in [9] as follows:

$$D_p(\theta,\omega) = \frac{\pi P(\theta,\omega)^2}{(1/2)\int_0^{2\pi} P(\theta,\omega)^2\, d\theta}, \tag{12}$$

where $P(\theta,\omega)$ is the output power of the array when pointing to the direction θ and $(1/2)\int_0^{2\pi} P(\theta,\omega)^2 d\theta$ is the sum of the squared 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. Consequently, D_p defines the quality of the microphone array and can be used to specify a certain threshold for the microphone array. For instance, if D_p equals 8, the main lobe is eight times slimmer than the unit circle and offers a confident estimation of a sound-source within half a quadrant.

Whereas D_p is usually considered for broadband sound sources, other metrics are necessary to profile the array's response for different types of sound sources. Figure 6 depicts the maximum side lobe (MSL) and the half-power beamwidth, which are two complementary metrics used to characterize the response of arrays for narrowband sound sources. Half-power beamwidth is the angular extent by which the power response has fallen to half of the maximum level of the main lobe. Since the half-power coincides with a 3 dB drop in power level, it is often called 3 dB beamwidth ($BW_{-3\,dB}$). This metric determines the angular ratio between the power signal level which is at least 50% of the peak power level and the remaining circle. By contrast, MSL is another important parameter used to represent the impact of the side lobes when characterizing arrays. MSL is the normalized ratio

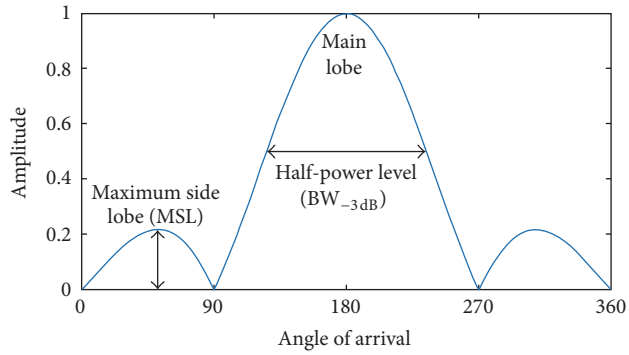

Figure 6: Definitions of maximum side lobe (MSL) and 3 dB beamwidth ($BW_{3\,dB}$).

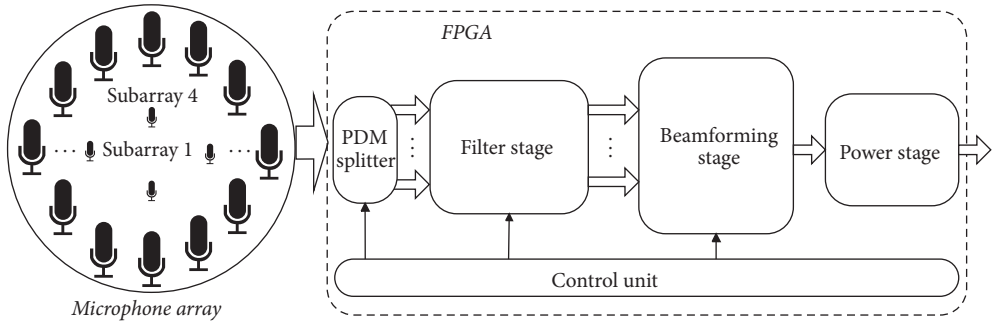

Figure 7: Main stages of the proposed architecture.

between the highest side lobe and the power level of the main lobe expressed in dB. Both metrics, the MSL and $BW_{-3\,dB}$, are desired to be as low as possible, whereas D_p should be as high as possible to guarantee a precise sound-source location.

4. A Filter-and-Sum Based Architecture

The proposed architecture uses a Filter-and-Sum based-beamforming technique to locate a sound-source with an array of digital MEMS microphones. Many applications, however, demand a certain scalability and flexibility when locating the sound-source. With such requirements in mind, the proposed architecture has some additional features to support a dynamic response targeting applications with real-time demands. The proposed architecture is also designed to be battery power efficient and to operate in streaming fashion to achieve the fastest possible response.

One of the features of the ADMP521 microphone is its low-power sleep mode capability. When no clock signal is provided, the ADMP521 microphone enters in a low-power sleep mode (<1 μA), which makes this sound-source localizer suitable for battery powered implementations. The PCB of the MEMs microphone array is designed to exploit this capability. Figure 2 depicts the subarray distribution of the MEMs microphones. Using the clock signal, it is possible to activate or deactivate subarrays since each subarray is fetched with an individual clock signal. This flexibility allows disabling not only subarrays of microphones, but also the associated computational components, decreasing the computational demand and the power consumption. The proposed architecture is properly designed to support such flexibility.

Table 1: Relevant parameters involved in proposed architecture.

Parameter	Definition
F_s	Sampling frequency
$F_{\min}$	Minimum frequency of the target sound source
$F_{\max}$	Maximum frequency of the target sound source
BW	Minimum bandwidth to satisfy Nyquist
D_F	Decimation factor
D_{CIC}	CIC filter decimation factor
N_{CIC}	Order of the CIC filter
D_{FIR}	FIR filter decimation factor
N_{FIR}	Order of the FIR filter

The array computes its response as fast as possible to reach real-time sound-source location. The proposed architecture is designed to process in stream fashion and is mainly composed of three cascaded stages operating in pipeline (Figure 7). The first stage is the filter chain, which is composed of the minimum number of components required to recover the audio signal in the target frequency range. The second stage computes the Filter-and-Sum beamforming operation. The final stage obtains $P(\theta, t)$ for the focused orientation. A *polar power map* is obtained once a complete steering loop is completed. The different stages are discussed in more detail in the following subsections. Table 1 summarizes the most relevant parameters of the proposed architecture.

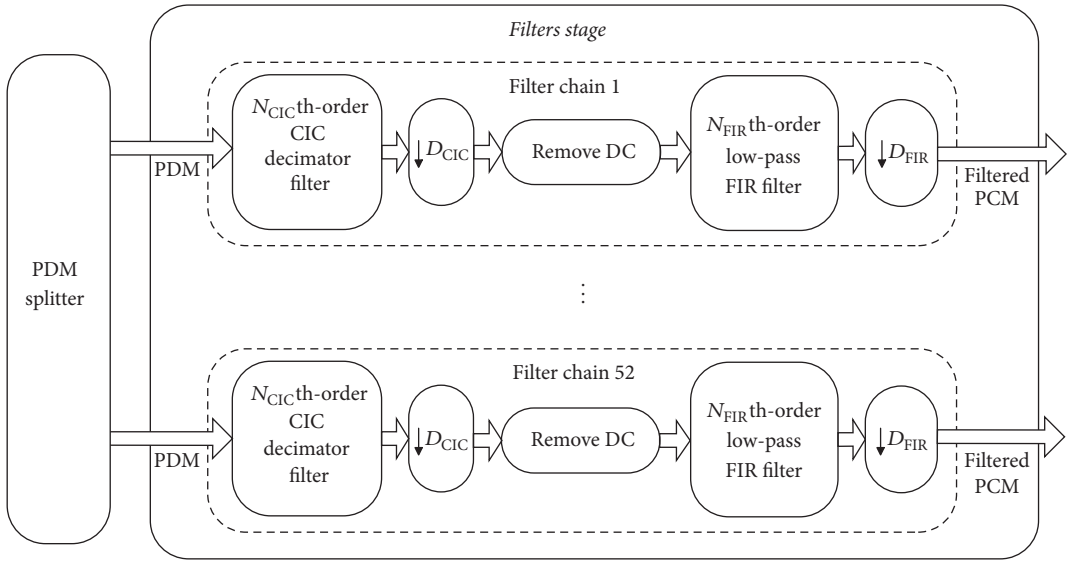

Figure 8: The filtering stage consists of a couple of filters with a downsampling factor.

4.1. Filter Stage. The filter stage contains a PDM demultiplexer and as many filter chain blocks as MEMS microphones (Figure 8). Each microphone of the array is associated with a filter chain composed of a couple of cascaded filters. The full-capacity design supports up to 52 filter chain blocks working in parallel, but their number is defined by the number of active microphones. The unnecessary filter chain blocks are disabled at runtime.

The microphones' clock F_S determines the input rate and, therefore, how fast the filter stage should operate. The low operating frequency for current FPGAs allows interesting power savings [17].

Every pair of microphones has its PDM output signal multiplexed in time. Thus, at every edge of the clock cycle the output is the sampled data from one of the microphones. The PDM demultiplexing is the first operation to obtain the individual sampled data from each microphone. This task is done in the PDM splitter block.

The next component consists of a cascade of filters to filter and to downsample each microphone signal. Traditional digital filters such as the Finite Impulse Response (FIR) type of filters are a good solution to reduce the signal bandwidth and to remove the higher frequency noise. Once the signal is filtered it can be decimated to decrease the oversampling to a reasonable audio quality rate (e.g., 48 kHz). However, this filter consumes many adders and dedicated multipliers (DSPs) from the FPGA resources, particularly if its order increases.

The Cascaded Integrated-Comb (CIC) filter is an alternative for low-pass filtering techniques which has been developed in [18, 19] and involves only additions and subtractions. This type of filter consists of 3 stages: the integrating stage, the decimator or integrator stage, and the comb section. PDM samples are recursively added in the integrating stage while being recursively subtracted with a differential delay in the comb stage. The number of recursive operations in both the integrating and comb section determines the order of the filter (N_{CIC}) and 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 (DD) and is exponential to the filter order [19].

$$G = \left(D_{CIC} \cdot DD\right)^{N_{CIC}}. \tag{13}$$

The output bit width grows proportionally to G. Denote by B_{in} the number of input bits; then the number of output bits B_{out} is as follows:

$$B_{out} = \left\lceil N_{CIC} \cdot \log_2\left(D_{CIC} \cdot DD\right) + B_{in} \right\rceil. \tag{14}$$

The proposed CIC decimation filter eliminates higher frequency noise components and decimates the signal by D_{CIC} at the same time. However, a major disadvantage of this filter is the nonflat frequency response in the desired audio frequency range. In order to improve the flatness of the frequency response, a CIC filter with a lower decimation factor followed by a compensation FIR filter is often chosen like in [20–22].

The CIC filter is followed by an averager, which is used to cancel out the effects caused by the microphones' DC offset output leading to a constant offset in the beamforming values. This block improves the dynamic range, reducing the bit width required to represent the data after the CIC.

The last component of each filter chain is a low-pass compensation FIR filter based on a Kaiser window. This filter equalises the passband drop usually introduced by CIC filters [19]. It additionally performs a low rate change. The proposed filter also needs 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}. This low-pass FIR filter is designed in a serial fashion to reduce the resource consumption. In fact, the FIR filter order

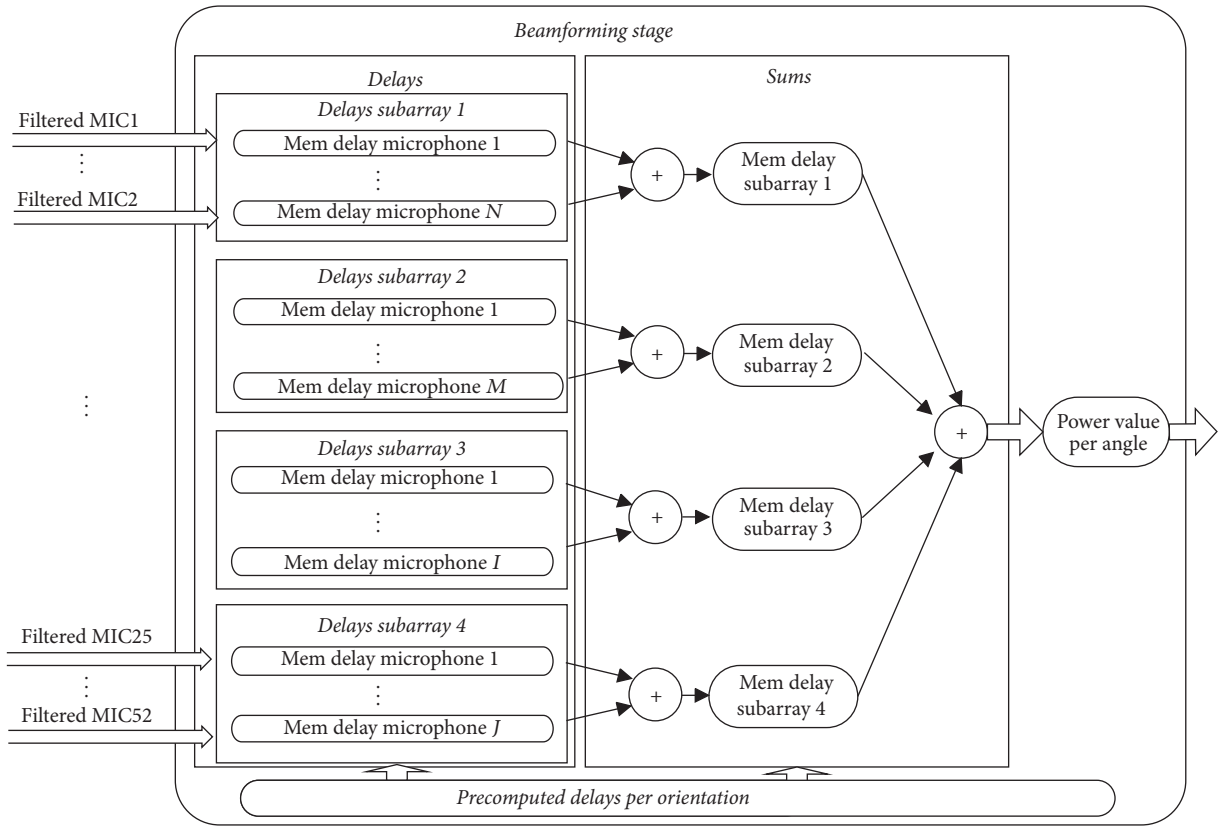

Figure 9: Details of the internal structure of the proposed modular Filter-and-Sum beamforming. Note that the delay values are stored in a precomputed table.

is also determined by D_{CIC}. Thereby the stream nature of the architecture, the CIC filter, is able to generate an output value every clock cycle. Due to the decimation factor, only one output value per D_{CIC} input value is propagated to the low-pass FIR filter. Therefore, the FIR filter has D_{CIC} clock cycles to compute each input value, which determines its maximum order. The filtered signal is then further decimated by a factor of D_{FIR} to obtain a minimum bandwidth BW $= 2 \cdot F_{\max}$ of audio signals to satisfy the Nyquist theorem. The overall D_F can be expressed based on the low rate change of each filter.

$$D_F = D_{\text{CIC}} \cdot D_{\text{FIR}}. \tag{15}$$

4.2. Beamforming Stage. As detailed before, the main purpose of the beamforming operation is to focus the MEMS microphone array in one particular direction. The detection of sound sources is possible by continuously steering in loops of 360°. The number of orientations, N_o, determines the angular resolution. 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 depends on the number of microphones and subarrays. Although Filter-and-Sum beamforming assumes a fixed number of microphones and a fixed geometry, our scalable solution satisfies those restrictions while offering a flexible geometry. Figure 9 shows our proposed Filter-and-Sum based beamformer. This stage is basically composed of FPGA's blocks of memory (BRAM) in ring-buffer fashion that properly delay the filtered microphone signal. The values of the delays at a given moment depend on the focus orientation at that moment and are determined by the array pattern $W(\omega_n, \theta_0, \theta)$ from (5). The delay for a given microphone is determined by its position on the array and on the focus orientation. All possible delay values per microphone for each beamed orientation are precomputed, grouped per orientation and stored in 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 precomputed table.

The beamforming stage is designed to support a variable number of microphones. This is enabled by grouping the input signals following their subarray structure. Therefore, instead of implementing one simple Filter-and-Sum of 52 microphones, there are four Filter-and-Sum operations in parallel for the 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 of the

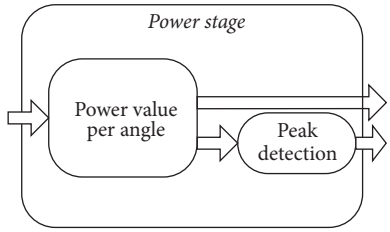

Figure 10: The power stage consists of a couple of components to calculate P-SRP and the estimated location of the sound-source.

subarrays. Although the output of some subarrays is already properly delayed, additional delays, shown at the Sums section in Figure 9, are inserted to assure that the proper delay of each subarray has been obtained. This is achieved by using the valid output signals of each subarray beamforming, without additional resource cost. Consequently, only the Filter-and-Sum beamforming modulo linked to an active subarray is enabled. The not active 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 reduction of the memory resource consumption. Since each subarray has their ring-buffer memory properly dimensioned to its maximum sample delay, the portion of underused regions of the consumed memories is significantly low.

4.3. Power Stage. Figure 10 shows the components of the power stage. Once the filtered data has been properly delayed and added for a particular orientation θ, $P(\theta, t)$ is calculated following (10). 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 direction shown by the peak of the *polar power map*, which corresponds to the orientation with the maximum $P(\theta, t)$.

5. Performance Analysis of the Filter-and-Sum Based Architecture

A performance analysis of the proposed architecture is presented in this section. The analysis shows how the design parameters such as the filters' characteristics affect the final execution time of the sound-source locator. The links between performance and design parameters are explained followed by the description of the different acceleration strategies. These strategies can be considered standalone or combined for certain timing constraints. The advantages of these strategies are lately presented in Section 6.

5.1. Time Parameters. The overall execution time of the proposed architecture is defined by the latency of the main components. A detailed analysis of the implementation of components and the latency that they incur provides a good insight about the speed of the system (Table 2). The operation frequency of the design can be assumed to be the same as the sampling frequency. Let us define $t_{\text{P-SRP}}$ as the overall execution time in clock cycles required to obtain P-SRP. Thus, $t_{\text{P-SRP}}$ is defined as

$$t_{\text{P-SRP}} = N_o \cdot t_o = N_o \cdot \left(t_{\text{filters}} + t_{\text{beamforming}} + t_{\text{power}}\right), \quad (16)$$

where t_o is the execution time of one orientation and is determined by the execution time of the filter stage (t_{filters}), the execution time of the beamforming ($t_{\text{beamforming}}$), and the execution time of the power stage (t_{power}), which are the main components of the system as explained in the previous section. The proposed architecture is designed to pipeline each stage, overlapping the execution of each component of the design. Therefore, only the initial latency or initiation interval (II) of the components needs to be considered, since it corresponds to the system group delay.

Let us assume that the design operates at the same frequency F_S like the microphones; then (16) can be rearranged as follows:

$$\begin{aligned} t_{\text{P-SRP}} &= \frac{N_o \cdot L_o}{F_S} \\ &= N_o \cdot \left(t_{\text{II}}^{\text{filters}} + t_{\text{II}}^{\text{beamforming}} + t_{\text{II}}^{\text{power}} + t_s\right), \end{aligned} \quad (17)$$

where L_o is the latency of the system and determined by the initiation interval of the filter stage ($t_{\text{II}}^{\text{filters}}$), the initiation interval of the beamforming stage ($t_{\text{II}}^{\text{beamforming}}$), and the initiation interval of the power stage ($t_{\text{II}}^{\text{power}}$). The time during which the microphone array is monitoring one particular orientation is known as t_s. This is the time required to calculate a certain number of output samples (N_s). As previously detailed, the digital microphones oversample the audio signal by operating at F_S. The reconstruction of the audio signal in the target range demands a certain level of decimation D_F.

Table 2: Relevant parameters involved in the performance calculation for the proposed architecture.

Parameter	Definition
t_s	Sensing time
t_o	Execution time of one orientation
N_o	Number of orientations
L_o	Latency of the system
$t_{\text{P-SRP}}$	Time required to obtain a *polar power map*
$t_{\text{II}}^{\text{filters}}$	Initiation interval of the filter stage
t_{filters}	Execution time of the filter stage
$t_{\text{II}}^{\text{beamforming}}$	Initiation interval of the beamforming stage
$t_{\text{beamforming}}$	Execution time of the beamforming stage
$t_{\text{II}}^{\text{power}}$	Initiation interval of the power stage
t_{power}	Execution time of the power stage
t_{II}	Sum of all initiation intervals
$t_{\text{II}}^{\text{CIC}}$	Initiation interval of the CIC filter
$t_{\text{II}}^{\text{DC}}$	Initiation interval of the removed DC block
$t_{\text{II}}^{\text{FIR}}$	Initiation interval of the FIR filter
$t_{\text{II}}^{\text{Delay}}$	Initiation interval of the delay memories
$t_{\text{II}}^{\text{Sum}}$	Initiation interval of the cascaded sums
$t_{\text{II}}^{\text{Power}}$	Initiation interval of the power calculation

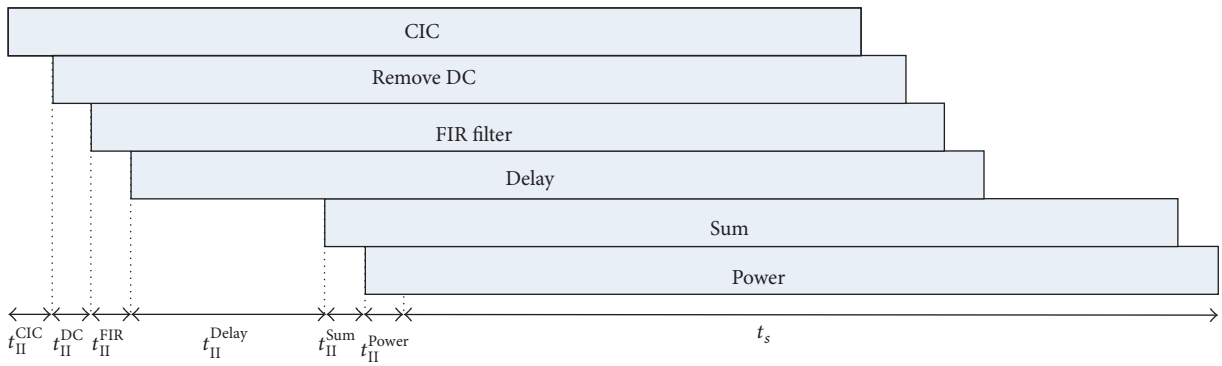

Figure 11: Timing analysis of the pipelined execution of the components.

This level of decimation is done by the CIC and the FIR filter in the filter stage, with a certain level of decimation (D_{CIC}) and (D_{FIR}), respectively. Based on D_F defined in (1), the time t_s is expressed as follows:

$$t_s = \frac{D_F \cdot N_s}{F_S} = \left\lceil \frac{F_S}{\text{BW}} \right\rceil \cdot \frac{N_s}{F_S} \approx \frac{N_s}{2 \cdot F_{\max}}. \tag{18}$$

II of each stage of the implementation can also be further decomposed based on the latency of the components,

$$\begin{aligned} t_{\text{II}}^{\text{filters}} &= t_{\text{II}}^{\text{CIC}} + t_{\text{II}}^{\text{DC}} + t_{\text{II}}^{\text{FIR}} \\ t_{\text{II}}^{\text{beamforming}} &= t_{\text{II}}^{\text{Delay}} + t_{\text{II}}^{\text{Sum}}, \end{aligned} \tag{19}$$

where t_{II}^{i} is the initiation interval of each component i. Therefore, t_{II} is defined as the sum of all the initiation intervals:

$$t_{\text{II}} = t_{\text{II}}^{\text{CIC}} + t_{\text{II}}^{\text{DC}} + t_{\text{II}}^{\text{FIR}} + t_{\text{II}}^{\text{Delay}} + t_{\text{II}}^{\text{Sum}} + t_{\text{II}}^{\text{Power}}. \tag{20}$$

Equation (16) can be rearranged (see Figure 11) as

$$t_{\text{P-SRP}} = N_o \cdot \left(t_{\text{II}} + t_s\right). \tag{21}$$

The execution time $t_{\text{P-SRP}}$ is determined by N_o and N_s, since the level of decimation is determined by the target frequency range and t_{II} is determined by the components' design. Although most of the latency of each component of the design is hidden thanks to the pipelined operation, there are still some cycles dedicated to initialize the components. A detailed analysis of t_{II} provides valuable information about the performance leaks.

CIC. The initiation interval of the CIC filter represents the time required to fulfil the integrator and the comb stages. Therefore, the order of the CIC (N_{CIC}) determines $t_{\text{II}}^{\text{CIC}}$.

$$t_{\text{II}}^{\text{CIC}} = \frac{2 \cdot N_{\text{CIC}} + 1}{F_S}. \tag{22}$$

DC. The component which must remove the DC level of the signal introduces a minor initial latency due to its internal registers. Since it needs at least two input values to calculate the DC level, it also depends on D_{CIC}.

$$t_{\text{II}}^{\text{DC}} = \frac{D_{\text{CIC}} + 2}{F_S}. \tag{23}$$

FIR. The initiation interval of the FIR filter is also determined by the order of this filter (N_{FIR}). Since the filter operation is basically a convolution, the initial output values are not correct until at least the $\lceil (N_{\text{FIR}} + 1)/2 \rceil$th input signal of the filter. Because the filters are cascaded, D_{CIC} also affects $t_{\text{II}}^{\text{FIR}}$.

$$t_{\text{II}}^{\text{FIR}} = \frac{D_{\text{CIC}} \cdot \left(\left\lceil \left(N_{\text{FIR}} + 1\right)/2 \right\rceil + 1\right)}{F_S}. \tag{24}$$

Therefore, $t_{\text{II}}^{\text{filters}}$ is expressed as follows:

$$\begin{aligned} t_{\text{II}}^{\text{filters}} &= t_{\text{II}}^{\text{CIC}} + t_{\text{II}}^{\text{DC}} + t_{\text{II}}^{\text{FIR}} \\ &= \frac{2 \cdot N_{\text{CIC}} + D_{\text{CIC}} \cdot \left(2 + \left\lceil \left(N_{\text{FIR}} + 1\right)/2 \right\rceil\right) + 3}{F_S}. \end{aligned} \tag{25}$$

Delay. The beamforming operation is done through memories, which properly delay the audio samples for a particular orientation. The maximum number of samples determines the minimum size of these delay memories. This value represents the maximum distance between a pair of microphones for a certain microphone array distribution and may vary for each orientation. The initiation interval of the Filter-and-Sum beamformer is therefore expressed as the maximum distance between pairs of microphones for a particular orientation.

$$t_{\text{II}}^{\text{Delay}} = \max\left(\Delta_{\text{am}}(\theta)\right) \cdot \frac{D_F}{F_S}, \tag{26}$$

where $\max(\Delta_{\text{am}}(\theta))$ is the maximum time delay of the active microphones for the beamed orientation θ. Therefore, $t_{\text{II}}^{\text{Delay}}$ is mainly determined by the microphone array distribution, F_S, and the target frequencies, determining D_F. Due to the symmetry of the microphone array and for the sake of simplicity, it is assumed that each orientation has the same $\max(\Delta_{\text{am}})$. Notice this does not need to be true for different array configurations.

Sum. The proposed beamforming is composed of not only a set of delay memories but also a sum tree. The initiation interval of this component is defined by the number of active microphones (N_{am}).

$$t_{\text{II}}^{\text{Sum}} = \frac{\lceil \log_2 (N_{\text{am}}) \rceil}{F_S}. \tag{27}$$

Therefore, $t_{\text{II}}^{\text{beamforming}}$ is expressed as follows:

$$\begin{aligned} t_{\text{II}}^{\text{beamforming}} &= t_{\text{II}}^{\text{Delay}} + t_{\text{II}}^{\text{Sum}} \\ &= \frac{\max (\Delta_{\text{am}} (\theta)) \cdot D_F + \lceil \log_2 (N_{\text{am}}) \rceil}{F_S}. \end{aligned} \tag{28}$$

Power. The final component is the calculation of the power per orientation. This simple component has a constant latency of a couple of clock cycles.

$$t_{\text{II}}^{\text{Power}} = \frac{2}{F_S}. \tag{29}$$

The timing analysis of the initiation interval of each component of the architecture gives an idea about the design parameters with higher impact. The definition of the filters, mainly their order, is determined by the application specifications, so it should not be modified to reduce the overall execution time. On the other hand, the distribution of the microphones in the array affects not only the frequency response of the system, but also the execution time. Notice, however, that the number of microphones does not have timing impact. Only the number of active microphones has a minor impact in terms of a couple of clock cycles of difference. Nevertheless, (21) already shows that the dominant parameters are t_s and N_o.

5.2. Sensitive Parameters. The timing analysis provides an indication of the parameters dominating the execution time. Some parameters, like the microphone array distribution, which determine the beamforming latency, are fixed while others like N_o or t_s per orientation are variable.

Orientations. Figure 5 depicts how an increment of N_o leads to a better sound-source localization. This resolution, however, has a high repercussion on the response time. A simple strategy is to maintain the angular resolution only for where it is needed while quickly exploring the surrounding sound field. For instance, the authors in [3] propose a strategy to reduce the beamforming exploration to 8 orientations, with an angular separation of 45 degrees. Once a steering loop ends, the orientations are rotated one position, which represents a shift operation in the precomputed orientation table. Therefore, all the supported 64 orientations are monitored after 8 steering loops. Despite this strategy intending to accelerate the peak detection by monitoring the minimum N_o, the overall N_o remains the same for achieving the equivalent angular resolution.

Sensing Time. The sensing time is a well-known parameter of radio frequency applications. The time t_s is known to strengthen the robustness against noise [23]. In our case, the time a receiver is monitoring the surrounding sound field determines the probability of properly detection of a sound-source. Consequently, a higher t_s is needed to detect and locate sound sources under low Signal-to-Noise (SNR) conditions. Despite the fact that this term could be modified in runtime to adapt the sensing of the array based on an estimated SNR, it would demand a continuous SNR estimation, which is out of the scope of this paper.

To conclude, Table 2 summarizes the timing definitions. On one hand, t_s determines the number of processed acoustic samples and therefore directly affects the sensing of the system. On the other hand, N_o determines the angular resolution of the sound-source search and influences the accuracy. There is a trade-off between t_s and N_o and the quality of the sound-source location.

5.3. Strategies for Time Reduction. The following three strategies are proposed to accelerate the sound-source localization without any impact on the frequency response and D_P of the architecture. An additional strategy is proposed specially for dynamic acoustic environments, but with a certain accuracy cost.

5.3.1. Continuous Processing. The proposed architecture is designed to reset the filter and beamforming stages after t_o due to orientation transition. Thanks to beamforming after the filter stage, the system can be continuously processing while resetting. The filter stage does not need to stop its processing. The input data is not lost due to the reset operations since the filtered input values are stored in the beamforming stage. Furthermore, the initialization of the beamforming stage can also be eliminated since the stored data from the previous orientation can be reused for the calculation of the new one. With this approach, (17) becomes as follows:

$$\begin{aligned} t_{\text{P-SRP}} &= t_{\text{II}}^{\text{filters}} + t_{\text{II}}^{\text{beamforming}} + N_o \cdot (t_{\text{II}}^{\text{power}} + t_s) \\ &\approx t_{\text{II}} + N_o \cdot t_s \end{aligned} \tag{30}$$

5.3.2. Time Multiplexing. Nowadays, FPGAs can operate at clock speeds of hundreds of MHz. Despite the fact that the power consumption is significantly lower when operating at low frequency [17], the proposed architecture is able to operate at much higher frequency than the data sampling rate. This capability provides the opportunity to parallelize the beamforming computations without any additional resource consumption. Instead of consuming more logic resources by replicating the main operations, the proposed strategy, similar to Time-Division Multiplexing in communications, consists in time multiplexing these parallel operations. Because the type of the input data is oversampled audio, the selection of the operations to be time multiplexed is limited. Based on (21), the candidates to be parallelized are N_o and t_s. Since the input data rate is determined by F_S, (18) shows that t_s cannot be reduced without decreasing N_s or changing the target frequency range. Nevertheless, since the computation of each orientation is data independent, they can be parallelized. The

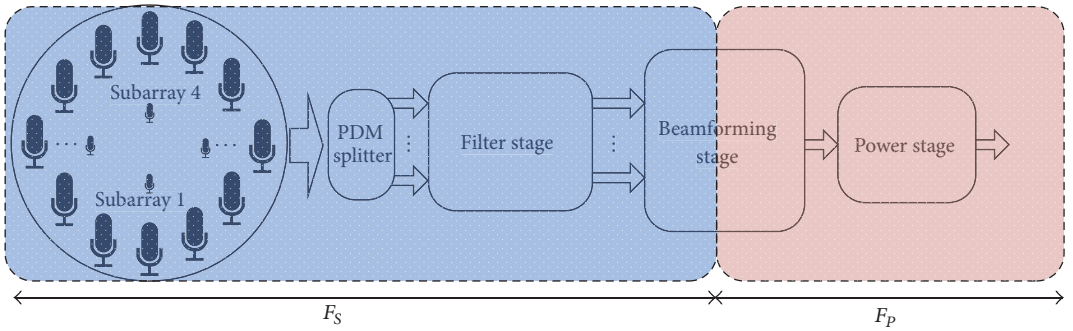

Figure 12: Clock regions for the time multiplexing of the computation of multiple N_o.

simultaneous computation of multiple orientations is only possible after the beamforming operation. Let us define t_{II}^{P} as the monitoring time before being able to process multiple orientations in parallel. Therefore,

$$t_{\text{II}}^{P} = t_{\text{II}}^{\text{CIC}} + t_{\text{II}}^{\text{DC}} + t_{\text{II}}^{\text{FIR}} + t_{\text{II}}^{\text{Delay}}. \tag{31}$$

After t_{II}^{P} the delay memories which compose the Filter-and-Sum beamforming stage have already stored enough audio data to start locating the sound-source. Because the beamforming operation relies on delaying the recovered audio signal, multiple orientations can be computed in parallel by accessing the content of the delay memories at a higher speed than the sampling of the input data. It basically multiplexes the output beamforming computations over time. The required frequency F_P to parallelize all N_o for this architecture is defined as follows:

$$F_P = F_S \cdot \frac{N_o}{D_F}. \tag{32}$$

Due to (1), F_P can be also expressed based on the target frequency range:

$$F_P \approx \text{BW} \cdot N_o. \tag{33}$$

Notice that the required frequency to multiplex in time the computation of the orientations does not depend on the number of microphones in the array. Figure 12 shows the clock domains when applying this strategy. While the front-end, consisting of the microphone array and the filter stage, operates at F_S, the output of the beamforming is processed at F_P. The additional cost in terms of resources is the extension of the register for the power per angle calculation. A memory of N_o positions is required instead of the single register used to store the accumulated power values. This strategy allows fully parallelizing the computation of all the orientations. Thus, $t_{\text{P-SRP}}$ is mainly limited by N_o and the maximum reachable frequency of the design, since F_S is determined by the microphones' operational frequency and D_F by the frequency range of the target sound-source. In fact, D_F determines how many orientations can be processed in parallel.

5.3.3. Parallel Time Multiplexing. This proposed strategy is an extension of the previous one. The frequency F_P is limited by the maximum attainable operating frequency of the implementation, which is determined by many factors, from the technology to the available resources on the FPGA. For instance, if $F_{\max}$ equals 30 kHz and the maximum attainable operating frequency is 100 MHz, then up to 1666 orientations could be computed in parallel. However, if not all the resources of the FPGA are completely consumed, especially the internal blocks of memory (BRAM), there is still space for improvement. With the time multiplexing strategy, the memories of the beamforming stage are fully accessed, since in each clock cycle there is at least one memory access or even two memory accesses when new data is stored. Therefore, more memory resources can be used to further accelerate the computation of the P-SRP. The simple replication of the beamforming stage, preconfigured for different orientations, will be enough to double the number of processed orientations while maintaining the same $t_{\text{P-SRP}}$. The strategy mainly consumes BRAMs. Nevertheless, due to the value of the $\max(\Delta_m)$ at BW for our microphone array, only few audio samples are needed to complete the beamforming. This fact drastically reduces the memory consumption, which provides the potential computation of thousands of orientations by applying both strategies.

All strategies can be applied independently despite the fact that some will only work properly when combined. Not all strategy combinations are beneficial. For instance, a dynamic angular resolution should be only combined with the time multiplexing of the orientations when F_P is higher than F_S. Otherwise the reduction of N_o by dynamically readjusting the target orientations does not provide any acceleration and it would only degrade the response of the system.

6. Results

The proposed architecture is evaluated in this section. Our analysis starts evaluating different design solutions based on the timing analysis introduced in Section 5.1. One representative configuration is evaluated based on the frequency response and accuracy by using the metrics described in Section 3.5. This evaluation also considers sensitive parameters such as the number of active subarrays and the relevance of N_o, already introduced in Section 5.2. The resource and the power consumption for a Zynq 7020 target FPGA are also

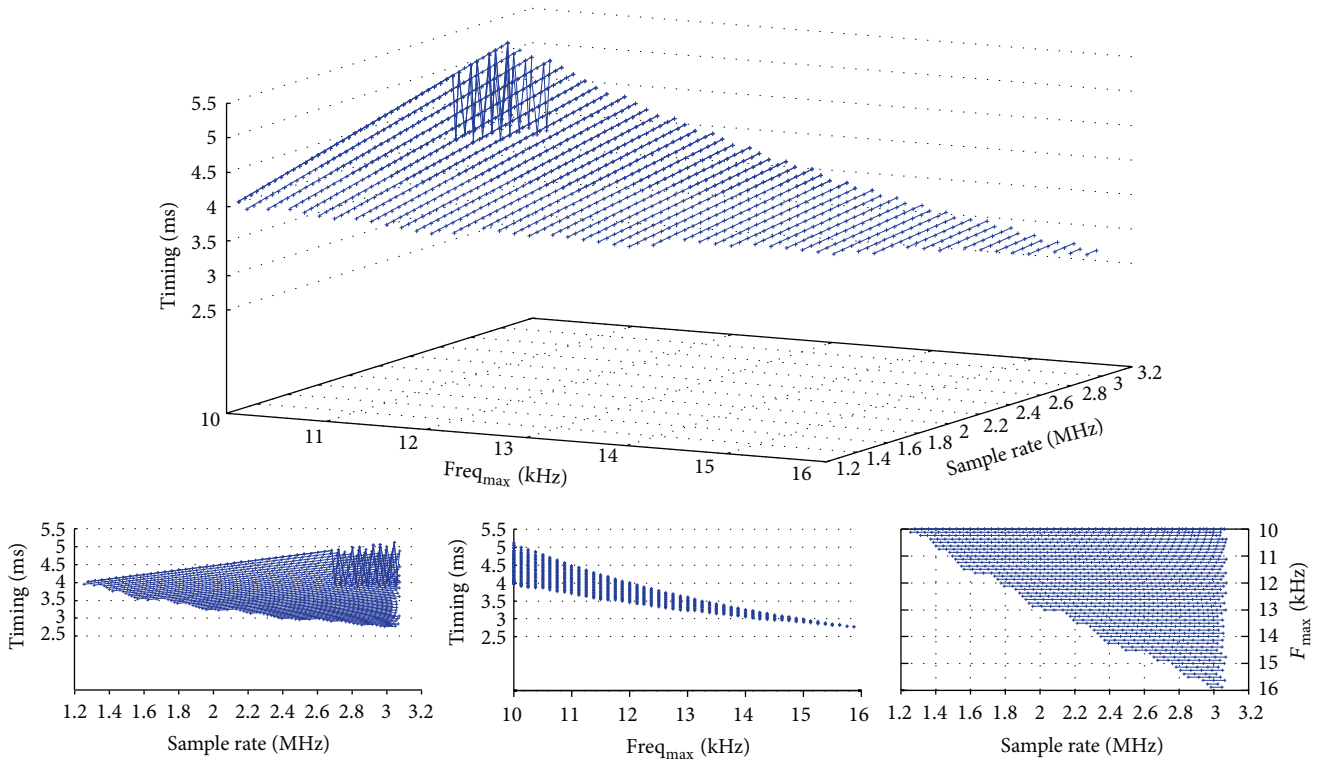

FIGURE 13: Minimum values of t_o based on F_S and F_{max}. Different perspectives are displayed in the bottom figures. Notice how the shortest t_o is obtained when increasing F_{max} and F_S.

presented. Finally, the strategies presented in Section 5.3 are applied for the representative design.

6.1. General Performance Analysis. The proposed performance analysis from the previous section is here applied on a concrete example. The explored design parameters are F_S and F_{max}, keeping N_s and N_o both constant to 64. Whereas F_S is determined by the microphone's sampling frequency, F_{max} is determined by the target application. For our design space exploration, we consider an F_{max} from 10 kHz to 16 kHz in steps of 125 Hz and F_S ranges from 1.25 MHz until 3.072 MHz as specified in [10].

Equations (16) to (18) and (20) to (32) are used to obtain $t_{P\text{-}SRP}$. The performance analysis starts obtaining D_F for every possible value of F_S and F_{max}. All possible combinations of D_{CIC} and D_{FIR} are considered based on (15). 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 2 kHz and 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. We consider these parameters as realistic constraints for low-pass FIR filters. Furthermore, a minimum order of 4 is defined as threshold for N_{FIR}. Thus, some values are discarded because D_F is a prime number or N_{FIR} is below 4. Each low-pass FIR filter is generated and evaluated in Matlab 2016b.

Figure 13 depicts the minimum timings of the DSE that the proposed Filter-and-Sum architecture needs to compute one orientation. t_o is slightly reduced when varying F_S. For instance, it is reduced from 5.03 ms to 3.97 ms when F_{max} = 10 kHz. A higher F_S means a faster sampling, which is in fact the operational frequency limiting factor. Furthermore, a higher decrement of $t_{P\text{-}SRP}$ is produced when increasing F_S and F_{max}. 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 while increasing F_{max}. Due to F_S and F_{max} ranges, the values of D_F vary between 39 and 154. Though, as previously explained, many values cannot be considered since they are either prime numbers or the decomposition in factors of D_{CIC} leads to values below 4. 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.

Finally, relatively low values of $t_{P\text{-}SRP}$ are obtained for F_{max} values from 10 kHz to 10.65 kHz and F_S ranging from 2.7 MHz to 3.072 MHz. It is produced by high values of D_{CIC}, which means that a higher order low-pass FIR filter is supported. As expected, high values of D_{CIC} lead to high order low-pass FIR filters and lower D_{FIR}. A lower $t_{P\text{-}SRP}$ is possible thanks to avoiding unnecessary computations since fewer samples are decimated after the low-pass FIR filter.

6.2. Analysis of a Design. As shown in Figure 13, several design considerations drastically affect the final performance.

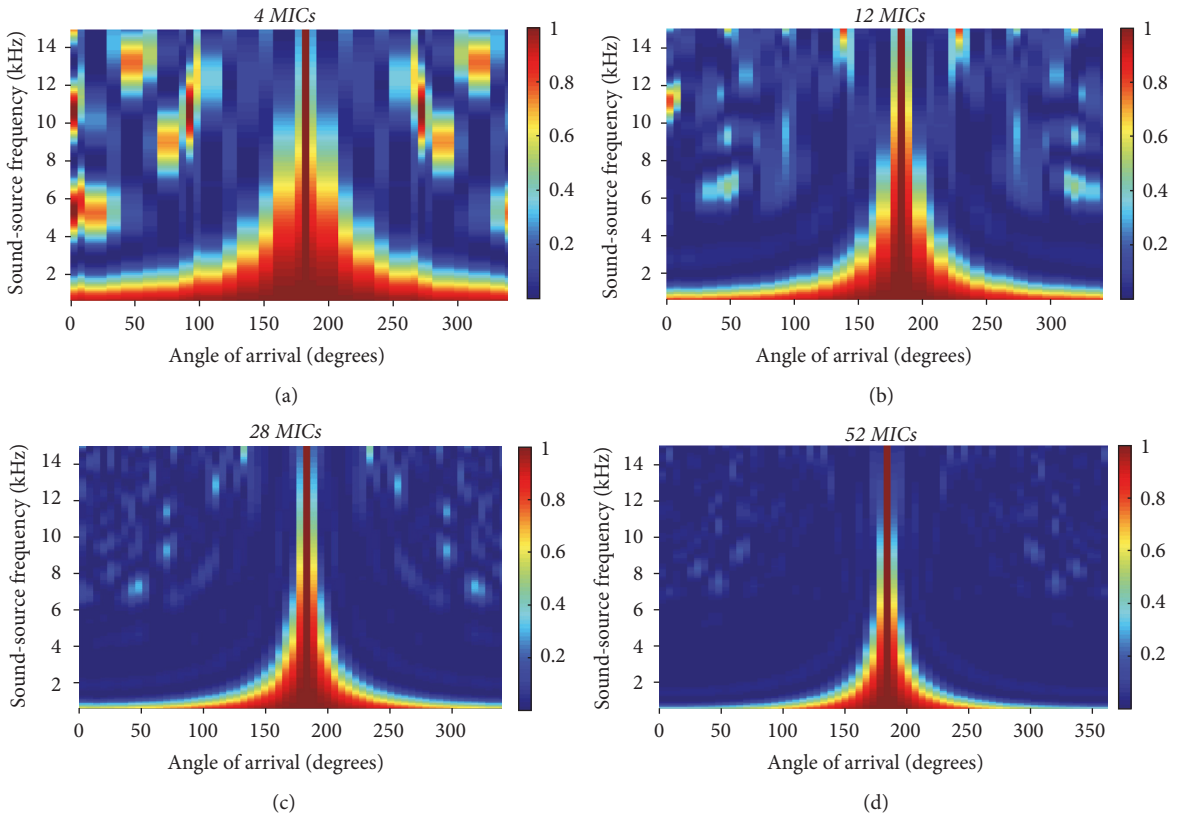

Figure 14: Waterfall diagrams of the proposed architecture. The figures are obtained by enabling only a certain number of subarrays. From (a) to (d): only the 4 innermost microphones, only the 12 innermost microphones, the 28 innermost microphones, and all microphones.

Table 3: Configuration of the architecture under analysis.

Parameter	Definition	Value
F_s	Sampling frequency	2 MHz
F_{min}	Minimum frequency	1 kHz
F_{max}	Maximum frequency	15.625 kHz
BW	Minimum bandwidth to satisfy Nyquist	31.25 kHz
D_F	Decimation factor	64
D_{CIC}	CIC filter decimation factor	16
N_{CIC}	Order of the CIC filter	2
D_{FIR}	FIR filter decimation factor	4
N_{FIR}	Order of the FIR filter	16

However, most of these design decisions do not have a significant impact on the system response compared to other factors such as the number of active microphones or the number of orientations. The analysis of impact of these parameters on the system's response and performance is done over one particular design.

Table 3 summarizes the configuration of the architecture. The design considers $F_s = 2$ MHz, which is the clock for the microphones and the functional frequency of the design. This value of F_s is the intermediate value between the required clock signals of the ADMP521 microphones [10]. The selected cut-off frequency is $F_{\text{max}} = 15.625$ kHz, which leads to $D_F = 64$. In this example design $N_{\text{CIC}} = 4$ with a decimation factor of 16 and a differential delay of 32. The chosen FIR filter has a beta factor of 2.7 and a cut-off frequency of F_{max} at a sampling rate of 125 kHz, which is the sampling rate obtained after the CIC decimator filter with a $D_{\text{CIC}} = 16$. The filtered signal is then further decimated by a factor $D_{\text{FIR}} = 4$ to obtain a BW = 31250 kHz audio signal.

The architecture is designed to support a complete steering loop up to 64 orientations, which represents an angular resolution of 5.625°. On the other hand, the subarray approach allows activating the 52 microphones if all the 4 subarrays are active. The final results are obtained by assuming a speed sound of ≈343.2 m/s.

6.2.1. Frequency Response. The waterfall diagrams of Figure 14 show the power output of the combined subarrays in all directions for all frequencies. In our case, the results are calculated with a single sound-source varying between 100 Hz and 15 kHz in steps of 100 Hz and placed at 180°. All results are normalized per frequency. Every waterfall shows a clear distinctive main lobe. When only subarray 1 is active there are side lobes at 5.3 kHz and 10.6 kHz which impede the sound-source location for those frequencies. The frequency response of the subarrays improves when they are combined since their frequency responses are superposed. The combination of the subarrays 1 and 2 reaches a minimum

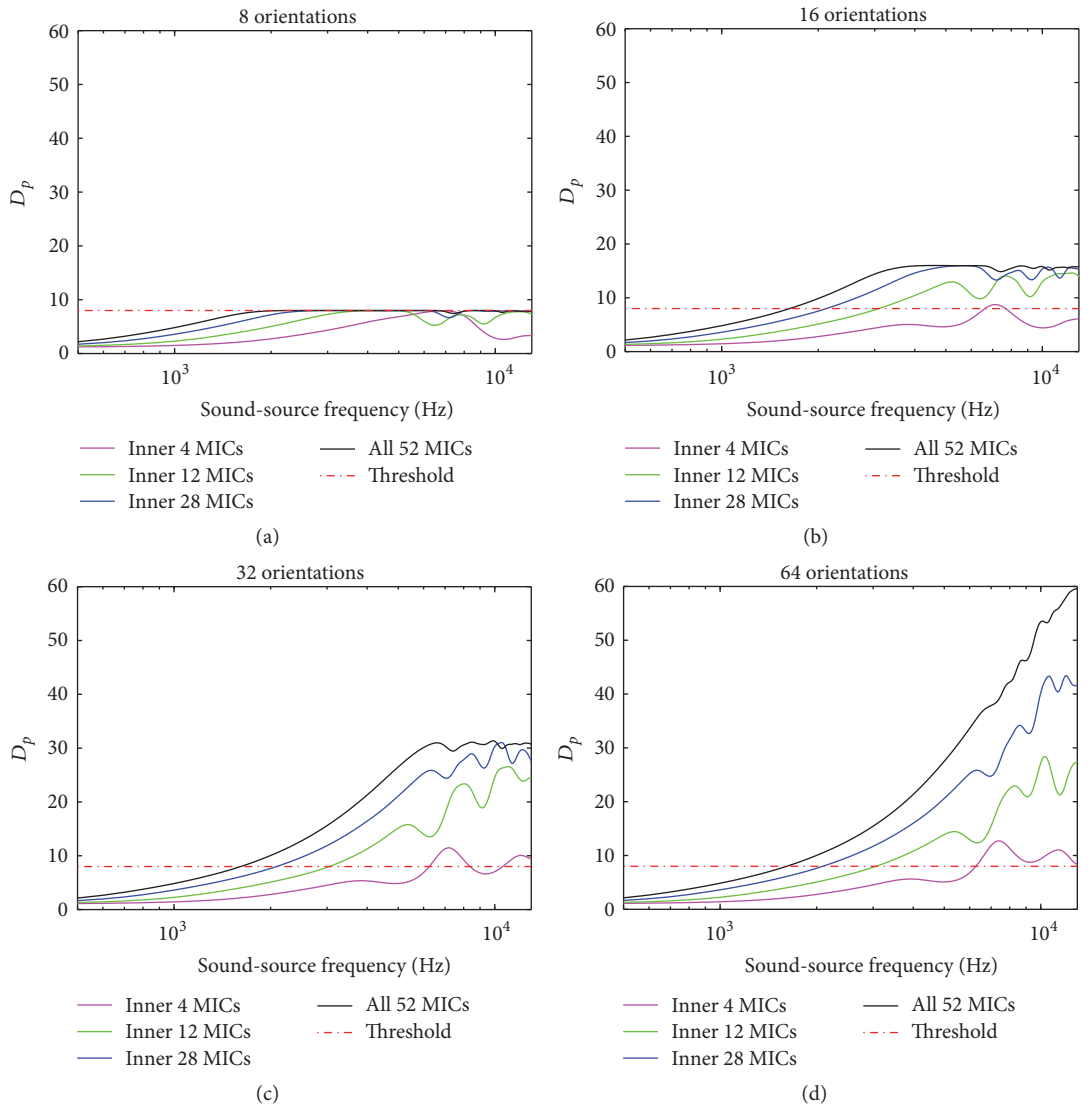

FIGURE 15: Directivities when considering a variable number of orientations and active microphones. From (a) to (d) D_P with only 8 orientations up to 64 orientations on (d).

detectable frequency of 3.1 kHz, when combining subarrays 1, 2, and 3 and all subarrays reach 2.1 kHz and 1.6 kHz, respectively. These minimum values are clearly depicted in Figure 15, with a threshold of 8 for D_P, which indicates that the main lobe's surface corresponds to maximally half of a quadrant. The frequency response of the combination of subarrays has a strong variation at the main lobe and, therefore, in D_P. Figure 15 depicts the evolution of D_P when increasing the angular resolution and when combining subarrays. The angular resolution determines that the upper bound D_P converges, which is dependent on the number of orientations. The number of active microphones, on the other hand, influences how fast D_P converges to its upper limit. Consequently, the number of active microphones determines the minimum frequency which can be located when considering a threshold of 8 for D_P. Alongside the directivity, other metrics such as the main beamwidth and the MSL levels metrics are also calculated to properly evaluate the quality of the array's response. Figure 16 depicts the MSL when varying the number of active subarrays and the number of orientations. A low angular resolution leads to a lower resolution of the waterfall diagrams, but only the metrics can show the impact. At frequencies between 1 and 3 kHz the main lobe converges to a unit circle, which can be explained by the lack of any side lobe. Higher frequencies present secondary lobes, especially when only the inner subarray is active, which increases the MSL values independently of the angular resolution. A low angular resolution leads to unexpected low values of MSL since the secondary lobes are not detected. On the other hand, a higher number of active microphones lead to lower values of MSL, independently of the angular resolution.

Figure 17 depicts the $\mathrm{BW}_{-3\,\mathrm{dB}}$ metric for a similar analysis of the number of microphones and angular resolution. On

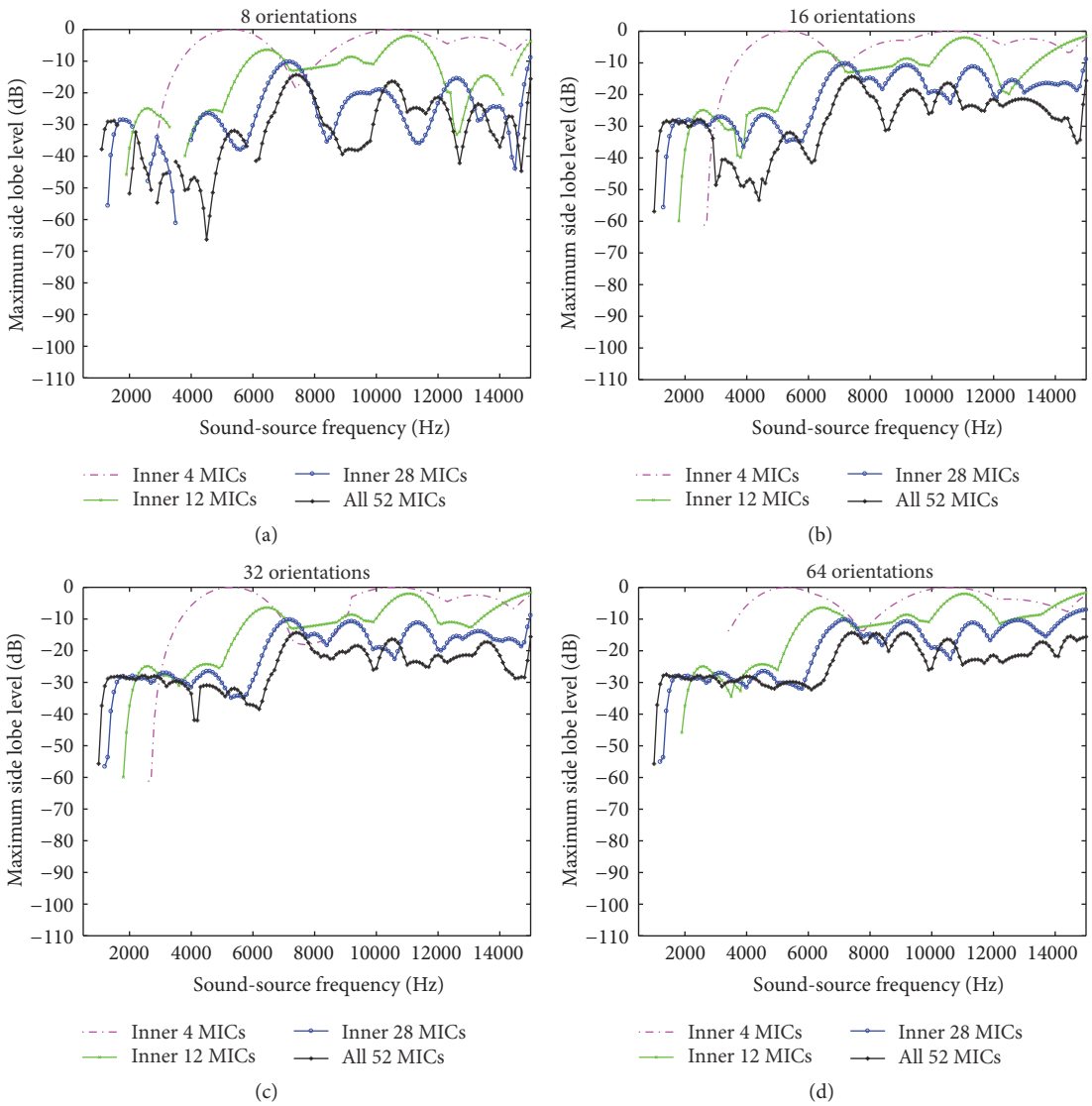

Figure 16: Measured MSL when considering a variable number of orientations and active microphones. From (a) to (d) the MSL with only 8 orientations up to 64 orientations on (d).

one hand, a higher number of microphones produce a faster decrement of $BW_{-3\,dB}$, reflected as a thinner main lobe. Nevertheless, $BW_{-3\,dB}$ of each subarray converges to a minimum, which is only reached at higher frequencies. The angular resolution determines this minimum, which ranges from 90° till 11.25° when 8 or 64 orientations are considered, respectively.

6.2.2. Resource Consumption and Power Analysis. Table 4 summarizes the resource consumption when combining subarrays. The consumed resources are divided into the resources for the filter stage, the beamforming stage, and the total consumption per groups of subarrays. The filter stage mostly consumes DSPs while the beamforming stage mainly demands BRAMs. Most of the resource consumption is dominated by the filter stage, since a filter chain is dedicated to each MEMs microphone. What determines the resource consumption is the number of active subarrays.

The flexibility of our architecture allows the creation of heterogeneous source-sound locators. Thus, the architecture can be scaled for small FPGAs based on the target sound-source profile or a particular desirable power consumption. For instance, the combination of the two inner subarrays would use 12 microphones while consuming less than 10% of the available resources. The LUTs are the limiting resource due to the internal registers of the filters. In fact, when all the subarrays are used around 80% of the available LUTs are required. Nevertheless, any subarray can be disabled in runtime, which directly deactivates its associated filter and beamforming components. Although this does not affect the resource consumption, it has a direct impact over the power consumption. Table 5 shows the power consumption

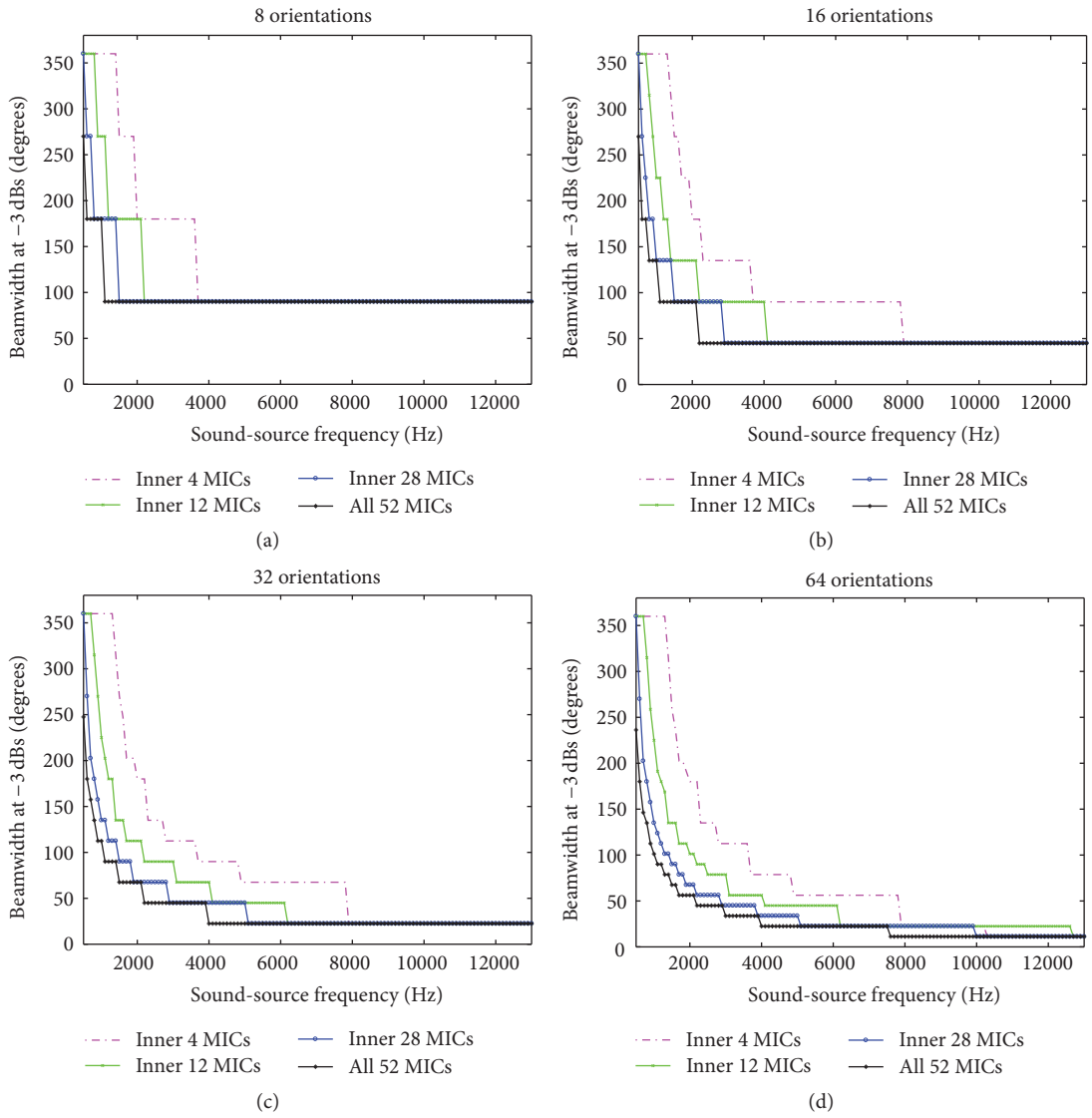

FIGURE 17: Measured $BW_{-3\,dB}$ level when considering a variable number of orientations and active microphones. From (a) to (d) the $BW_{-3\,dB}$ with only 8 orientations up to 64 orientations on (d).

in mW based on the number of active subarrays. The power consumption of the microphones is also considered since the FPGA and the microphone array are powered from the same source. Thus, the overall power consumption must be considered since the architecture is designed for an embedded system. The MEMS microphones are powered with 3.3 volts, which represents a power consumption per microphone of 2.64 μW and 3.96 mW for the inactive and active microphones, respectively. Notice how the power consumption increases with the number of active subarrays. There is a turning point when 3 or 4 subarrays are active. Thus, the microphone array consumes more power than the FPGA when all the subarrays are active.

6.2.3. Timing Analysis. The timing analysis based on Section 5 of the design under evaluation is summarized in Table 6. A complete steering loop requires around 169 ms while t_o rounds to 2.6 ms. Notice that the initialization (t_{II}) consumes around 21.5% of the execution time. Fortunately, this initialization can almost be completely removed when applying the first strategy described in Section 5.3.1.

Table 7 summarizes the timing results when applying the first strategies proposed in Section 5. The elimination of the initialization after each orientation's transition slightly reduces $t_{\text{P-SRP}}$. In this case, $t_{\text{P-SRP}}$ is expressed as follows:

$$t_{\text{P-SRP}} = t_{\text{II}} + N_o \cdot t_s. \tag{34}$$

The main improvement is obtained after time multiplexing the computation of the power per orientations. In this case, F_P, the operational frequency of the beamforming computation to process all N_o in parallel equals F_S as expressed in (32). This is possible because D_F and N_o have the same value. Therefore, there is no need to have a different clock for the beamforming operation, since the spacing between

Table 4: Resource consumption after placement and routing when combining microphone subarrays. Each subarray combination details the resource consumption of the filter and the beamforming stage.

Resources	Available	Inner 4 MICs			Inner 12 MICs			Inner 28 MICs			All 52 MICs		
	Resources	Filters	Beamforming	Total	Filters	Beamforming	Total	Filters	Beamforming	Total	Filters	Beamforming	Total
Slice registers	106400	5043	626	6144	14859	1540	16882	34489	3195	38183	54042	4447	59093
Slice LUTs	53200	3612	344	4732	10759	754	12299	25032	1486	27318	37221	2221	42319
LUT-FF	86689	2329	199	2773	7013	512	7779	16353	1069	17698	23656	1664	27619
BRAM	140	0	2	2	0	6	6	0	14	14	0	22	22
DSP48	220	8	4	12	24	4	28	56	4	60	88	4	92

Table 5: Power consumption at $F_s = 2$ MHz expressed in mW when combining microphone subarrays. Values obtained from the Vivado 2016.4 power report.

Active	MEMS microphones			Reported on-chip power			Total
Subarrays	Active	Inactive	Total	Static	Dynamic	Total	Power
Inner 4 MICs	15.84	0.13	15.97	120	2	122	137.97
Inner 12 MICs	47.52	0.11	47.63	120	5	125	172.63
Inner 28 MICs	110.88	0.06	110.94	121	11	132	242.94
All 52 MICs	205.92	0	205.92	122	16	138	343.92

Table 6: Timing analysis without any optimization of the design under evaluation. The values are expressed in μs.

Parameter	Definition	Values [μs]
$t_{\text{II}}^{\text{CIC}}$	Initiation interval of the CIC filter	4.5
$t_{\text{II}}^{\text{DC}}$	Initiation interval of the removed DC block	9
$t_{\text{II}}^{\text{FIR}}$	Initiation interval of the FIR filter	72
$t_{\text{II}}^{\text{Delay}}$	Initiation interval of the delay memories	480
$t_{\text{II}}^{\text{Sum}}$	Initiation interval of the cascaded sums	3.5
$t_{\text{II}}^{\text{Power}}$	Initiation interval of the power calculation	1
$t_{\text{II}}^{\text{filters}}$	Initiation interval of the filter stage	85.5
$t_{\text{II}}^{\text{beamforming}}$	Initiation interval of the beamforming stage	484.5
$t_{\text{II}}^{\text{power}}$	Initiation interval of the power stage	1
t_{II}	Sum of all initiation intervals	571
t_s	Sensing time	2048
t_o	Execution time of one orientation	2650
$t_{\text{P-SRP}}$	Time required to obtain a *polar power map*	169600

output filtered values from the filter stage is large enough. By combining the first two strategies, $t_{\text{P-SRP}}$ rounds to 2 ms and only the first steering loop needs 2.6 ms due to t_{II}^{P}. In this case, $t_{\text{P-SRP}}$ is expressed as follows:

$$t_{\text{P-SRP}} = t_{\text{II}}^{P} + t_s \approx t_s. \tag{35}$$

The other two strategies proposed in Section 5.3.1 are designed to fully exploit the FPGA resources and to overcome time constraints when considering a high angular resolution. In the first case, since the design under evaluation has a small angular resolution ($N_o = 64$), there is no need for a higher F_P when applying the time multiplexing strategy. However, a higher angular resolution can be obtained when considering the unconsumed resources without additional timing cost. Table 8 shows the combination of strategies increases the angular resolution without additional time penalty. The operational frequency (F_{op}) determines at what speed the FPGA can operate. By following (33), the beamforming operation can be exploited by increasing F_P up to the maximum frequency, which increases N_o as well:

$$\max\left(N_o\right) = \frac{\max\left(F_{\text{op}}\right)}{\text{BW}} = \frac{F_P}{\text{BW}}. \tag{36}$$

Many thousands of orientations can be computed in parallel when combining all strategies. The beamforming stage can be replicated as many times as the remaining available resources allow. Of course, this estimation is certainly optimistic since the frequency drops when the resource consumption increases. Nevertheless, this provides an upper bound for N_o. For instance, when only the inner subarray is considered, the DSPs are the limiting component. However, up to 53 beamforming stages could be theoretically placed in parallel. When more subarrays are active the BRAMs are the constrained component. Notice how the number of supported orientations increases if the number of subarrays decreases. It has, however, an impact on the frequency response and the accuracy of the system, as shown in Section 6.2.1. Nevertheless, tens of thousands of orientations can be computed in parallel consuming only around 2 ms by operating at the highest F_{op} and by replicating the beamforming stage to exploit all the available resources.

Table 7: Timing analysis of the optimized designs when applying and combining the first two strategies. The values are expressed in ms.

	Initial	Continuous	Time multiplexing	Continuous time multiplexing
$t_{\text{P-SRP}}$	169.6 ms	131.6 ms	2.6 ms	2 ms

Table 8: Maximum N_o when combining strategies. The maximum number of beamformers is obtained based on the available resources and the resource consumption of each beamformer (Table 4). The maximum F_{op} is reported by the Vivado 2016.4 tool after placement and routing.

	Continuous time multiplexing				Parallel continuous time multiplexing			
	Inner 4 MICs	Inner 12 MICs	Inner 28 MICs	All 52 MICs	Inner 4 MICs	Inner 12 MICs	Inner 28 MICs	All 52 MICs
max beamformers	—	—	—	—	55	23	10	6
max F_{op}	95.62 MHz	93.27 MHz	91.97 MHz	87.91 MHz	95.62 MHz	93.27 MHz	91.97 MHz	87.91 MHz
max N_o	3059	2984	2943	2813	168292	68650	29430	16879

7. Conclusions

In this paper we have presented a scalable and flexible architecture for fast sound-source localization. On one hand, the architecture can flexibly disable sections of the microphone array that are not needed or disable them to respect power restrictions. The modular approach of the architecture allows scaling the system for a larger or smaller number of microphones. Nevertheless, such capabilities do not impact the frequency and accuracy of our sound-source locator. On the other hand, several strategies to offer real-time sound-source localization have been presented and evaluated. These strategies not only accelerate but also provide solutions for those time stringent applications with a high angular resolution demand. Thousands of angles can be monitored in parallel, offering a high-resolution sound-source localization in a couple of milliseconds.

Conflicts of Interest

The authors declare that there are no conflicts of interest 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] E. Zwyssig, M. Lincoln, and S. Renals, "A digital microphone array for distant speech recognition," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP '10)*, pp. 5106–5109, IEEE, Dallas, Tex, USA, March 2010.

[2] A. Abdeen and R. Laxmi, "Design and performance of a real-time acoustic beamforming system," in *Proceedings of the 12th SENSORS '13 Conference*, IEEE, Baltimore, MD, USA, November 2013.

[3] B. da Silva, L. Segers, A. Braeken, and A. Touhafi, "Runtime reconfigurable beamforming architecture for real-time sound-source localization," in *Proceedings of the 26th International Conference on Field-Programmable Logic and Applications (FPL '16)*, IEEE, Lausanne, Switzerland, September 2016.

[4] Y. Zhang and S. Baobin, "Sound source localization algorithm based on wearable acoustic counter-sniper systems," in *Proceedings of the 5th International Conference on Instrumentation and Measurement, Computer, Communication, and Control, IMCCC '15*, pp. 340–345, IEEE, Qinhuangdao, China, September 2015.

[5] J. Sallai, W. Hedgecock, P. Volgyesi, A. Nadas, G. Balogh, and A. Ledeczi, "Weapon classification and shooter localization using distributed multichannel acoustic sensors," *Journal of Systems Architecture*, vol. 57, no. 10, pp. 869–885, 2011.

[6] T. Inoue, R. Imai, Y. Ikeda, and Y. Oikawa, Hat-type hearing system using MEMS microphone array, 2016.

[7] Z. I. Skordilis, A. Tsiami, P. Maragos, G. Potamianos, L. Spelgatti, and R. Sannino, "Multichannel speech enhancement using MEMS microphones," in *Proceedings of the 40th International Conference on Acoustics, Speech, and Signal Processing, ICASSP '15*, pp. 2729–2733, IEEE, Brisbane, Australia, April 2014.

[8] I. Salom, V. Celebic, M. Milanovic, D. Todorovic, and J. Prezelj, "An implementation of beamforming algorithm on FPGA platform with digital microphone array," in *Proceedings of the 138th Audio Engineering Society Convention, AES '15*, Audio Engineering Society, New York, Ny, USA, May 2015.

[9] 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.

[10] Analog Devices, "ADMP521 datasheetUltralow noise microphone with bottom Port and PDM digital output," Technical Report, Analog Devices, Norwood, MA, USA, 2012.

[11] Texas Instruments, "How delta-sigma ADCs work," Tehcnical report, Texas Intruments, http://www.ti.com/lit/an/slyt423/slyt423.pdf.

[12] D. H. Johnson and D. E. Dudgeon, *Array Signal Processing: Concepts and Techniques*, Simon & Schuster, New York, NY, USA, 1992.

[13] J. J. Christensen and J. Hald, "Technical Review Beamforming," Tech. Rep., Bruel & Kjear, Danmark, 2004.

[14] J. H. DiBiase, *A high-accuracy, low-latency technique for talker localization in reverberant environments using microphone*

arrays [Phd thesis], Brown University, Providence, RI, USA, 2000.

[15] J. H. DiBiase, H. F. Silverman, and M. S. Brandstein, "Robust localization in reverberant rooms," in *Microphone Arrays*, pp. 157–180, Springer, Berlin, Germany, 2001.

[16] M. J. Taghizadeh, P. N. Garner, and H. Bourlard, "Microphone array beampattern characterization for hands-free speech applications," in *Proceedings of the 7th Sensor Array and Multichannel Signal Processing Workshop, SAM '12*, pp. 465–468, IEEE, Hoboken, NJ, USA, June 2012.

[17] H. Blasinski, F. Amiel, and E. Thomas, "Impact of different power reduction techniques at architectural level on modern FPGAs," in *Proceedings of the Latin American Symposium on Circuits and Systems LASCAS*, Stanford University, Stanford, Calif, USA, 2010.

[18] 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.

[19] M. P. Donadio, "CIC filter introduction," 2000, https://pdfs.semanticscholar.org/5bf7/48fbdeb1ff68a2407c0ccfd58b816e9937d5.pdf.

[20] N. Hegde, "Seamlessly interfacing MEMs microphones with blackfin processors," EE-350 Engineer-to-Engineer Note, 2010.

[21] G. J. Dolecek and J. Diaz-Carmona, *On Design of CIC Decimators*, INTECH Open Access Publisher, 2011.

[22] R. Lyons, "Understanding cascaded integrator-comb filters," *Embed System Program*, vol. 18, no. 4, pp. 14–27, 2005.

[23] T. E. Bogale, L. Vandendorpe, and L. L. Bao, "Sensing throughput tradeoff for cognitive radio networks with noise variance uncertainty," in *Proceedings of the 9th International Conference on Cognitive Radio Oriented Wireless Networks, CROWNCOM '14*, pp. 435–441, IEEE, Oulu, Finland, June 2014.