

A Low-Power FPGA-Based Architecture for Microphone Arrays in Wireless Sensor Networks

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

Vrije Universiteit Brussel (VUB),
INDI department, Brussels, Belgium
`{bruno.da.silva}@vub.be`

Abstract. Microphone arrays add an extra dimension to sensory information from Wireless Sensor Networks by determining the direction of the sound instead of only its intensity. Microphone arrays, however, need to be flexible enough to adapt their characteristics to realistic acoustic environments, while being power efficient, as they are battery-powered. Consequently, there is a clear need to design adaptable microphone array nodes enabling quality aware distributed sensing and prioritizing low power consumption. In this paper a novel dynamic, scalable and energy-efficient FPGA-based architecture is presented. The proposed architecture applies the Delay-and-Sum beamforming technique to the single-bit digital audio from the MEMS microphones to obtain the relative sound power in the time domain. As a result, the resource consumption is drastically reduced, making the proposed architecture suitable for low-power Flash-based FPGAs. In fact, the architecture's power consumption estimation can become as low as $649 \mu\text{W}$ per microphone.

1 Introduction

Microphone arrays composed of Micro-Electro Mechanical systems (MEMS) microphones are becoming popular as they are now also applied as nodes in Wireless Sensor Networks (WSNs). This is possible due to their relatively low cost and high level of integration. For instance, they have been used to automatically emphasize the speech coming from a particular direction [1] or for urban environmental monitoring [2], [3]. Many applications benefit from the use of microphone arrays since they not only promise audio enhancement but also allow to determine the sound's Direction-of-Arrival (DoA). However, most of these applications need an accurate sound-source localization, which often can not be done with a standalone array. Existing solutions propose WSNs composed of microphone arrays for sensing the acoustic environment, locally processing the measured information and propagating it through a network to combine multiple captures. Despite the importance of power consumption in battery-based WSN nodes, it is often not considered.

Microphone arrays have been used as distributed acoustic sensing nodes for a broad range of applications. Sound-source detection using WSNs is usually related to surveillance, acoustic enhancement urban environmental monitoring

or military applications. For instance, the authors in [6] and in [7] propose WSN counter-sniper systems composed of microphone arrays. Whereas the first one uses Wi-Fi as wireless communication, Bluetooth is proposed in the last one. None of the solutions, however, report their power consumption.

The authors in [8] propose a Wi-Fi based WSN composed of microphone arrays for deforestation detection. Their architecture computes the audio from an array composed of 8 microphones in an extremely low-power Flash-based FPGA, which allows to only consume 21.8 mW per node in the network. The same authors propose in [9] a larger microphone array composed of 16 microphones. Due to the additional computational operations, they consider a Xilinx Spartan6 FPGA. The power consumption, however, increases up to 61.71 mW for the 16 microphones' configuration. Our proposed architecture, instead, allows to compute more than 4 times the number of microphones with 6 times less power consumption thanks to a reduced resource requirements.

Other technologies also provide low the power solutions. A very interesting solution is proposed in [10], where the authors present a very low-power microphone array. Their architecture only consumes 1.8 mW per microphone thanks to exploiting the sleep modes of the microcontroller and microphones. The microphones are inactive 20% of the time and the microcontroller is only active during the I^2C communication.

An acoustic sensor called SoundCompass, capable of measuring sound intensity and directionality, has been developed in [3] to satisfy the requirements of sound-source localization applications. The SoundCompass is composed of digital MEMS microphone arrays, designed to function in a distributed manner as part of a WSN or as standalone node. A WSN composed of SoundCompasses is not only able to sample the sound field directionality, but also to fuse this information for applications such as sound-source localization or real-time noise maps. The original SoundCompass, however, lacks a good time response, is not power efficient, and does not offer a dynamic response to spontaneous acoustic events critical for many applications. New architectures have been recently proposed to increment the dynamism. For instance, the architecture in [4] is designed to perform a fast and power-efficient sound-source location by dynamically adapting both the number of beamed orientations and microphones. The architecture is based on a variant of the Filter-and-Sum beamforming, implementing a filter stage for each microphone before computing the beamforming operation. This architecture request many FPGA resources, leading to a relative high power consumption ranging from 122 to 138 mW [5]. In this paper we present an architecture prioritizing the power consumption by drastically reducing the resource consumption while maintaining the scalability and dynamism presented in previous architectures. A minimal resource-greedy architecture will require a totally different approach, which is presented in the following section. To the best of our knowledge, the new architecture achieves the lowest power per microphone ratio compared to existing solutions.

The low-power architecture is described and evaluated in Section 2 and in Section 3 respectively. The conclusions are drawn in Section 4.

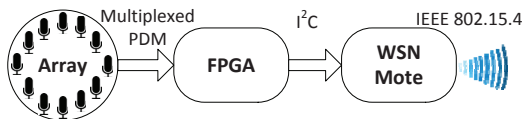


Fig. 1: Overview of the proposed WSN node. The proposed architecture computes the acquired audio signal to fully perform the sound source localization on the FPGA.

2 Architecture Description

Our proposed architecture for locating sound-sources in a 1kHz to 15kHz range is fully implemented in an FPGA, integrating the beamforming of the input signal, the filtering and audio conversion, and the sound's DoA. The architecture remains completely scalable and dynamic to adapt its response to the acoustic environment or to certain constraints such as extreme low-power conditions. The active configuration is received through the WSN mote. As a result, the architecture allows to activate or deactivate multiple microphones or to change the number of beamed orientations at runtime while continuing the processing as proposed in [4].

The following sections describe the sensor array, the FPGA's components and the WSN interface of a standalone device. The network analysis and considerations when combining multiple microphone arrays in a WSN are out of the scope of this paper and have been partially covered in [11].

2.1 Microphone Array

The proposed architecture for WSN relies on the same microphone array planar geometry as [3], where 52 MEMS microphones are placed on a 20 cm diameter planar geometry and grouped in four concentric sub-arrays of 4, 8, 16 and 24 MEMS microphones (Figure 2). The circular distribution of the microphones intends to maintain the array's response independent of the orientation. Each sub-array is differently positioned in order to facilitate the capture of the spatial acoustic information to be used by a beamforming technique for the localization of the sound source. The number of active microphones has a direct impact on the array's output signal-to-noise ratio (SNR) since it increases with the number of active microphones.

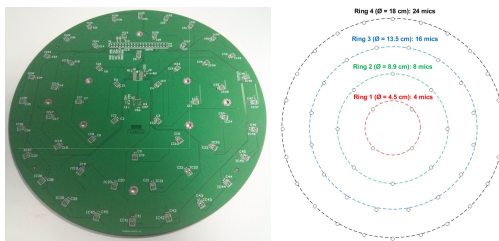


Fig. 2: The 52 digital MEMS microphones are distributed in 4 concentric circular MEMS microphone sub-arrays that can be activated or deactivated in runtime.

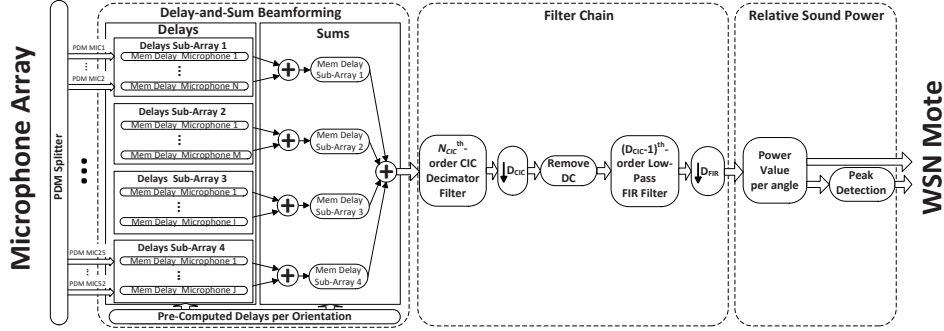


Fig. 3: Overview of the FPGA's components. The Delay-and-Sum beamforming is composed of several memories to properly delay the input signal. Our implementation groups the memories associated to each sub-array to disable those memories linked to deactivated microphones. The beamformed input signal is converted to audio in the cascade of filters. The DoA is finally obtained based on the relative sound power obtained per orientation.

The microphones selected to compose the microphone array are digital MEMS microphones with a multiplexed pulse density modulation (PDM) output. Nowadays digital MEMS microphones such as the ICS-41350 from InvenSense [14] provide good omnidirectional polar response, a wide-band frequency response ranging from 100 Hz up to 15 kHz and offer a low-power sleep mode which drastically reduces the power consumption. The deactivation of the microphone's clock signal activates this low-power sleep mode. From the other side, the digital MEMS microphones need a clock in a 1 to 3 MHz range to oversample the audio signal by a factor of 64. The PDM signal needs to be filtered to remove the high-frequency noise and to be downsampled to retrieve the audio signal in a Pulse-Code Modulation (PCM) format.

2.2 FPGA

Figure 3 depicts the main components of the FPGA's implementation. The input rate is determined by the microphone's clock, which corresponds to the sampling frequency (F_s). The oversampled PDM signal coming from the microphones is multiplexed per microphone pair. A PDM splitter block demultiplexes this signal at every edge of the clock cycle and splits the sampled PDM into 2 PDM separate channels. The obtained PDM streams from each microphone of the array are properly delayed to perform the beamforming operation, called Delay-and-Sum. This beamforming technique allows to amplify the sound coming from the set direction while suppressing the sound coming from other directions. Several cascaded filters remove the high-frequency noise and downsample the input signal to retrieve the audio signal. Finally, a *polar steering response map*, whose lobes are used to estimate the DoA for the localization of sound sources, is generated from the relative sound power.

To achieve the highest response time, this implementation is designed to operate in streaming mode, which warranties that each component is always computed after an initial latency.

Delay-and-Sum Beamforming The beamforming stage is composed by a bank of memories, a pre-computed table of delays and cascaded additions (Figure 1). The bank of memories is used to delay the different digital audio streams for the beamforming algorithm. Every microphone m is associated to a memory, which properly delays that particular audio stream with an amount Δ_m . The delay memories are grouped based on sub-arrays. Each delay memory belonging to a sub-array has the same width and length to support all the possible orientations. The width is determined by the PDM representation, which only needs one bit to represent the audio signal. The length is defined by the maximum delay ($\max(\Delta_i)$) of that sub-array i , which is determined by the MEMS microphone planar distribution and F_s . In fact, the maximum $\max(\Delta_i)$ determines the overall latency of the beamforming operation. Once the PDM input data is properly delayed for a particular orientation, the outputs of each memory are all added. This results in a summed PDM stream of the delayed PDM signals from the microphones.

Filters Description The oversampled PDM signals from the digital MEMS microphones need to be downsampled and filtered to retrieve the original acquired audio signal. The downsampling is done by a cascade of a CIC decimator filter and a low-pass FIR filter. The CIC filter is an alteration on the FIR filter for which no multiplications are required, becoming less computationally intensive and less resource greedy [12]. Thus, a CIC filter with a N_{CIC} order, a decimation factor of D_{CIC} and a differential delay DD is chosen in our design based on the selected F_s . The CIC filter is followed by a signal averaging block to cancel out the effects caused by the microphones' DC offset output, improving the dynamic range and reducing the bit width required to represent the data after the CIC. The last cascaded filter is a low-pass compensation FIR filter designed in a serial fashion to reduce the resource consumption. Consequently, the maximum order (N_{FIR}) of the low-pass FIR filter is determined by D_{CIC} . The filtered signal is then further decimated by a factor of D_{FIR} to obtain the minimum bandwidth BW to satisfy the Nyquist theorem.

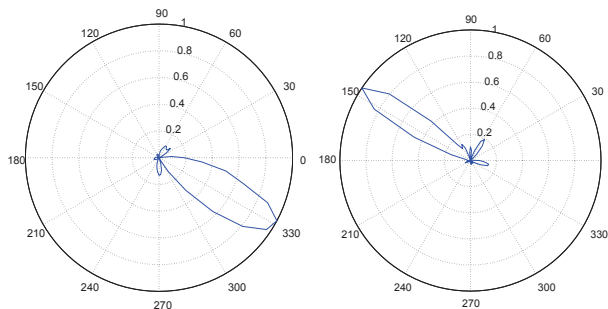


Fig. 4: Examples of P-SRP depicting the output power obtained under experimental conditions for sound sources of 3kHz (left figure) and 5kHz (right figure).

Relative Sound Power 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 *Polar Steered Response Power* (*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 known as sensing time (t_s). This is a well-known parameter on radio frequency applications, which is known to increment the robustness against the noise. A higher t_s is needed to detect and locate sound sources under low signal-to-noise (SNR) conditions. All the power signals in one steering loop conform the *P-SRP* (Figure 4). The peaks identified in the *P-SRP* point to the potential presence of sound sources.

The *P-SRP* is usually calculated in the frequency domain [3], using the Fourier transform, which increases the resource consumption and potentially enlarges the time the system focuses on a particular direction. In our architecture, the power of the signal is obtained in the time domain by applying the Parseval's theorem.

2.3 Wireless Sensor Network Mote

The proposed architecture includes a wireless communication capability. The calculation of *P-SRP* is performed in the FPGA, while the wireless communication is done externally by a low-power WSN mote. Figure 5 depicts the selected device, a Zolertia WSN platform Z1 based on the MSP430F2617 microcontroller [15]. This WSN mote is chosen due to its flexibility since it supports several wireless technologies such as IEEE 802.15.4 and 6LoWPAN. Another interesting feature of this mote is its low-power consumption, being on average 40 mW.

The communication between the FPGA and the Zolertia mote is done through an Inter-Integrated Circuit (I^2C), which is a serial communication bus system. I^2C uses a serial data line and a serial clock line to interconnect the FPGA and the Zolertia mote. It supports an extremely wide clock frequency range, reaching up to 400 Kb/s, enough to transmit the *P-SRP* values or to receive the configuration control signals to determine the number of active microphones or the number of orientations from the network.

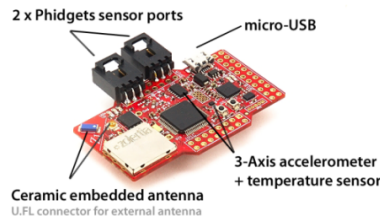


Fig. 5: The Zolertia WSN mote provides the wireless capability needed for our microphone array.

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
DD	CIC Differential Delay	32
D_{CIC}	CIC Filter Decimation Factor	32
N_{CIC}	Order of the CIC Filter	4
D_{FIR}	FIR Filter Decimation Factor	2
N_{FIR}	Order of the FIR Filter	31

Table 1: Configuration of the architecture under analysis.

3 Design Analysis

In this section, the proposed architecture is firstly compared to the one presented in [5], discussing the frequency response, resource and power consumption and the time performance. The section concludes with a comparison with state-of-the-art related architectures.

The configurations of the architecture under evaluation are summarized in Table 1. The variation of the target F_{Max} and the F_s directly affects to the beamforming stage by determining the length of the memories, and to the filter stage, by determining the decimation factor and the FIR Filter order. Although, the impact of the number of active microphones, which changes in runtime thanks to the sub-array distribution, is also analysed. The impact of the number of orientations is not evaluated here since it is partially discussed in [5]. For our evaluation, a complete steering loop is composed of 64 orientations, which represents an angular resolution of 5.625° .

3.1 Frequency Response

The frequency response of the microphone array is determined by the number of active microphones. Our experiments cover four configurations with 52, 28, 12 or 4 microphones determined by the number of active sub-arrays.

The proposed architecture is evaluated for three configurations (Table 1) by utilizing the directivity (D_P) to properly evaluate the quality of the array's response. The directivity reflects the ratio between the main lobe's surface and the total circle. Here we consider a threshold of 8 for D_P , which indicates that the main lobe's surface corresponds to at maximum half of a quadrant. The directivity is evaluated by placing a sound source at the 64 supported orientations. The average of all directivities along with the 95% confidence interval is calculated for the supported orientations. Figure 6 (left) depicts the resulting directivities based on the active sub-arrays for the proposed architecture. In case the 4 inner microphones are enabled, the directivity in all directions does not reach the predefined ratio of 8. When 12 microphones are enabled the directivity increases, and reaches the value of 8 at 3.1 kHz. This value is reached at 2.1 kHz and 1.7 kHz when 28 and all microphones are enabled. One can also note that the

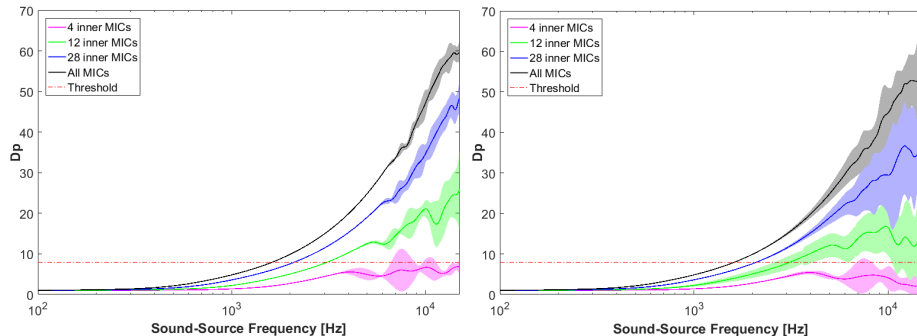


Fig. 6: Average D_P with a 95% confidence interval for the supported orientations when combining sub-arrays of the proposed architecture (left) and the architecture presented in [5] (right).

95 % confidence noticeably increases at 4 kHz, 6 kHz and 7 kHz for respectively the inner 4, 12 and 28, and all microphones.

The proposed architecture outperforms the frequency response of the architecture in [5], which is depicted in Figure 6 (right). The variance of D_P of the architecture in [5] increases with the sound source frequency, becoming very sensitive to the beamed orientation. The proposed architecture has higher beamforming resolution thanks to beamforming before downsampling the input data. Instead, the architecture in [5] performs the beamforming after the filter stage, whose data has a lower rate. Nevertheless, as shown in Figure 6, the capacity of properly determining the DoA increases with the number of active microphones. The price to pay, however, is a higher resource and power consumption as detailed below.

3.2 Resource Consumption

The proposed architecture drastically reduces the resource consumption. Table 2 details the resource consumption when targeting a Zynq 7020 FPGA. Although the low resource consumption of this architecture allows to use a smaller and lower demanding power FPGA, the Zynq 7020 FPGA is used in order to fairly compare this new architecture with the one presented in [4] and accelerated in [5]. The amount of different types of resources demanded by the proposed architecture is significantly lower than the architecture presented in [4],[5]. The reduction of the 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 in [4], [5] each microphone has an individual filter chain, the proposed architecture only needs one. The percentage of resources dedicated to the filter chains represents around 91% of the registers and 89% for LUTs in [5]. This percentage decreases to 14.7% and 32.8% of the consumed registers and LUTs respectively in the proposed architecture. An efficient memory partition is possible thanks to the storage of PDM signals and to the use of LUTs as internal memory. Despite the proposed architecture is also constrained by the number

Resource	Available Resources	Inner 4 MICs		Inner 12 MICs		Inner 28 MICs		Inner 52 MICs	
		[4], [5]	Proposed	[4], [5]	Proposed	[4], [5]	Proposed	[4], [5]	Proposed
Registers	106400	6144	1381	16882	1529	38183	1892	59093	2425
LUTs	53200	4732	1224	12299	1361	25032	2471	42319	4117
BRAM18k	140	2	1	6	1	14	1	22	1
DSP48	220	12	6	28	6	60	6	92	6

Table 2: Zynq 7020 resource consumption after placement and routing when combining microphone sub-arrays.

Active Sub-Arrays	MEMS Microphones			Reported On-Chip Power			WSN Mote	Total Power
	Active	Deactive	Total	Static	Dynamic	Total		
Inner 4 MICs	1.332	0.576	1.908	16.323	0	16.323	40	58.231
Inner 12 MICs	3.996	0.480	4.476	16.323	0	16.323	40	60.799
Inner 28 MICs	9.324	0.288	9.612	16.327	0.074	16.401	40	66.013
All 52 MICs	17.316	0	17.316	16.327	0.086	16.413	40	73.729

Table 3: Power consumption expressed in mW when combining microphone sub-arrays of a WSN node, including the microphones, FPGA and WSN mote power consumption. Values are obtained from the Libero SoC v.11.8 power report for the FPGA operating at $F_s = 2.08$ MHz, considering the low-power mode of the microphones [14] and [15].

of available LUTs, their consumption is much lower, allowing to use LUTs for internal memory of the beamforming stage. This is not beneficial in [5] because LUTs are the constraint resource, increasing the consumption of BRAMs. As a result, the larger configuration of the proposed architecture demands up to 24 times less registers and 10 times less LUTs. In fact, the available resources in the Zynq 7020 allow up to 10 instantiations of this architecture, which represents the computation of more than 500 microphones simultaneously.

3.3 Power Analysis

The low resource requirements of the proposed architecture allows to target low-power FPGAs. Flash-based FPGAs like Microsemi’s Igloo2, PolarFire or SmartFusion2 offer not only 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 [13].

The proposed architecture has been evaluated for a SmartFusion2 M2S050 (Table 3). The reported power consumption rounds to 16.4 mW, which represents a significant reduction compared to the one reported in [5], ranging from 122 mW to 138 mW. Nevertheless, notice that the target FPGA in that case is a Zynq 7020. Our architecture presents a major reduction of the power consumption when compared to [5], achieving the lowest power per microphone ratio when all the sub-arrays are active.

3.4 Timing Analysis

The execution time (t_{P-SRP}) on the proposed architecture is the time needed to obtain the $P-SRP$. This time is distributed between the computation of three main operations: beamforming, filtering and resetting. The memories, which are

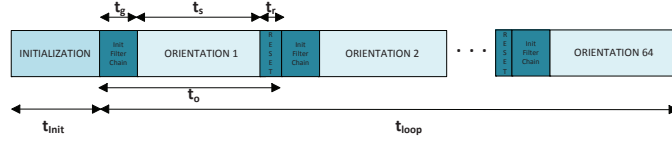
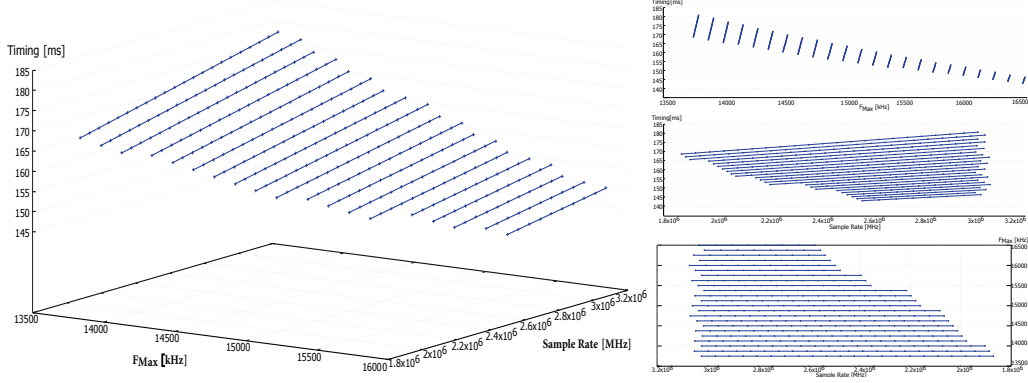


Fig. 7: Detailed schedule of the operations computed in serial.

Fig. 8: Minimum t_{P-SRP} when evaluating values of F_{Max} and F_s . Different perspectives are displayed in the right side.

composing the Delay-and-Sum beamforming implementation, need to be fetched with the input PDM samples before starting the filtering and the calculation of the P -SRP. This initial time (t_{Init}) is constant, since it depends on the microphones planar distribution, and rounds to 500μ .

The time needed per orientation (t_o) is determined by the sensing time t_s , the group delay of the filter stage (t_g), and the time to reset the filters (t_r) at the end of the computation of each orientation. The time t_g groups the initiation interval (II) needed by the block in the filter stage before generating a valid output. This time depends on the filters characteristics, detailed in Table 1, and has a

Parameter	Definition	Equation	Value [cc/MHz]
t_{II}^{CIC}	II of the CIC Filter	$2 \cdot N_{CIC} + 1$	$9/F_s = 4.33 \mu s$
t_{II}^{DC}	II of the Remove DC	$D_{CIC} + 2$	$34/F_s = 16.35 \mu s$
t_{II}^{FIR}	II of the FIR Filter	$D_{CIC}^2/2 + 1$	$513/F_s = 246.6 \mu s$
t_{II}^{Delay}	II of the delay memories at F_s	$max(\Delta)$	$1023/F_s = 491.18 \mu s$
t_{II}^{Sum}	II of the cascaded sums	$2 \cdot \lceil \log_2(N_{am}) \rceil$	$12/F_s = 5.77 \mu s$
t_s	Sensing Time	$D_{CIC} \cdot D_{FIR} \cdot (N_s - 1)$	$4032/F_s = 1.94 ms$
t_g	Group Delay	$t_{II}^{CIC} + t_{II}^{DC} + t_{II}^{FIR}$	$556/F_s = 267.31 \mu s$
t_o	Time per Orientation	$t_g + t_s$	$4588/F_s = 2.2ms$
t_{init}	II of the Delay-and-Sum	$t_{II}^{Delay} + t_{II}^{Sum}$	$1035/F_s = 497.59 \mu s$
t_{P-SRP}	Time to obtain a complete P-SRP	$t_{init} + N_o \cdot t_o$	$294667/F_s = 141.66 ms$

Table 4: 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.

References	Device	Mic	Time [ms]	Time/Mic [ms/Mic]	Power [mW]	Power/Mic [mW/Mic]
[8]	Igloo 2	8	-	-	30.44	3.80
[9]	Spartan 6	16	18.85	1.18	78.99	4.94
[10]	EFM32	4	249	62.25	7.2	1.8
[5]	Zynq 7020	52	2	0.04	343.92	6.61
Proposed	SmartFusion2	52	141.66	2.724	33.78	0.65

Table 5: Comparison of the time performance and the reported power consumption of the microphone array and the FPGA.

significant impact on the time performance. The time t_o can be approximated to:

$$t_o = t_s + t_g + t_r \approx t_s + t_g \quad (1)$$

because only few cc are needed to reset the filters. The execution time to obtain P -SRP (t_{P-SRP}) as detailed in Figure 7 is:

$$t_{P-SRP} = t_{init} + N_o \times t_o = t_{init} + t_{loop} \quad (2)$$

where N_o is the number of orientations, t_{init} is the initialization time of the beamforming operation, t_o is the time one orientation needs to be computed and t_{loop} is the time to compute N_o orientations. The t_{P-SRP} for the analyzed equals to 141 ms .

Table 4 provide further details about the timing analysis and includes the equations for the timing analysis, which are determined by the architecture design. Figure 8 shows a design space exploration similar to the one done in [5]. The architecture is evaluated for F_{max} ranging from 10 kHz to 16.5 kHz in steps of 125 Hz and F_s ranging from 1.25 MHz until 3.072 MHz. The order of the FIR filter (N_{FIR}) and the decimations factors D_{CIC} and D_{FIR} are obtained based on F_s and F_{max} . The equations in Table 4 are used to obtain t_{P-SRP} for each design. The frequency range of the target application determines F_{min} and F_{max} , which is used to select the F_s that offers the highest time performance in the proposed architecture. Unfortunately, due to the redesign of the architecture, the strategies like a faster clock proposed in [5] cannot be applied without a significant increment of the resource consumption.

3.5 Comparison

Table 5 summarizes the comparison of the proposed 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. Despite the proposed architecture is substantially slower than the one presented in [5], the time-per-microphone ratio is better than other related solutions.

4 Conclusions

The proposed architecture demonstrates that large MEMS microphone arrays are suitable for WSN, even when they are composed of tens of MEMS microphones. The drastic reduction of the resource requirements allows to consider

more power efficient devices such as flash-based FPGAs. The price to pay is an acceptable degradation in the time response. Nevertheless, the new architecture not only offers a better frequency response but also an interesting balance between time performance and power consumption for applications on WSN.

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 20142020 through the ERDF-2020 Project ICITYRDI.BRU.

References

1. Zwysig, E., et al. "A digital microphone array for distant speech recognition." Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on. IEEE, 2010.
2. Zhang, X., et al. "Design of small MEMS microphone array systems for direction finding of outdoors moving vehicles." Sensors 14(3) : 4384-4398. 2014.
3. Tiete, J., et al. "SoundCompass: a distributed MEMS microphone array-based sensor for sound source localization". Sensors, 14(2), 1918-1949. 2014.
4. da Silva, B., et al. "Runtime reconfigurable beamforming architecture for real-time sound-source localization." Field Programmable Logic and Applications (FPL), 2016 26th International Conference on. EPFL, 2016.
5. da Silva, B., et al. "Design Considerations when Accelerating an FPGA-Based Digital Microphone Array for Sound-Source Localization." Journal of Sensors (2017):2
6. Ledeczi, A., et al. "Countersniper system for urban warfare." ACM Transactions on Sensor Networks (TOSN) 1.2 (2005): 153-177. 2005.
7. Sallai, J., et al. "Weapon classification and shooter localization using distributed multichannel acoustic sensors." Journal of Systems Architecture 57.10 (2011): 869-885. 2011.
8. Petrica, L., et al. "Energy-Efficient WSN Architecture for Illegal Deforestation Detection." Int J Sensors Sensor Netw 3.3: 24-30. 2015.
9. Petrica, L. "An evaluation of low-power microphone array sound source localization for deforestation detection." Applied Acoustics 113: 162-169. 2016.
10. Ottoy, G., et al. "A low-power MEMS microphone array for wireless acoustic sensors." Sensors Applications Symposium (SAS), 2016 IEEE. IEEE, 2016.
11. da Silva, Bruno, et al. "A partial reconfiguration based microphone array network emulator." Field Programmable Logic and Applications (FPL), 2017 27th International Conference on. IEEE, 2017.
12. Hogenauer, E. "An economical class of digital filters for decimation and interpolation." Acoustics, Speech and Signal Processing, IEEE Transactions on 29(2): 155-162. 1981.
13. Microsemi, User Guide 0444 V5 (UG0444), *SmartFusion2 SoC and IGLOO2 FPGA Low-Power Design*, available online, 2017
14. InvenSens. ICS-41350 datasheet, 2017.
15. Zolertia WSN platform, Z1 Datasheet, Mar. 2010