Area and Power Optimised ASIC Implementation of Adaptive Beamformer for Hearing Aids

Kartik Samtani, Jobin Thomas, Deepu S. P., Sumam David S., SMIEEE
Department of Electronics and Communication Engineering
National Institute of Technology Karnataka, Surathkal, India
Email: samtanikm@gmail.com

***Abstract*—Beamforming is a technique used in hearing aids to improve the intelligibility of target sound by reducing the interference from other directions. An efficient ASIC implementation of a two omnidirectional microphone array based adaptive beamforming algorithm is presented in this paper with various optimisations proposed at different stages of the hardware design. The beamform patterns and improvements in SNR values obtained from experiments conducted in a conference room environment were analyzed to verify the working of the design. The architecture was implemented with 0.18 μm standard cell libraries using Cadence Design Tools. Cell area and power reports were analysed for different optimisations and area and power obtained are 0.054 mm^2 and 60.54 μW respectively.**

***Index Terms*—Hearing Aids; Beamforming; Adaptive Filtering; Real Time Hardware Design; Noise Reduction**

I. Introduction

Most of the methods used for improving Signal to Noise Ratio (SNR) of speech signal in hearing aids rely on subtracting noise spectrum from the input signal in time or frequency domain. But in challenging environments where the target speaker's voice gets affected by other speaker's speech or speech-like signals, these methods do not perform well since it is difficult to identify and separate such noise effectively from the required signal. Beamforming, a spatial filtering technique, performs better in such scenarios, in which sound from the direction of the target speaker is received with high sensitivity while attenuating the interference from other directions.

Beamforming algorithms are based on processing the signals recorded by an array of omnidirectional microphones placed at particular distance and obtaining required directional patterns by superimposing these signals. There are mainly two approaches for directional noise cancellation, either have fixed beamform patterns [1], [2] or have adaptively varying beamform patterns [3], [4]. Adaptive beamforming gives good improvements in the SNR values as the beamform patterns change with the direction of noise source. Beamforming algorithms available in the literature are mostly implemented on general purpose Digital Signal Processors [4].

In this paper, we have done an area and power optimized ASIC implementation of a two microphone array based adaptive beamforming technique for hearing aids. Various hardware optimisations are followed to reduce the power consumption of the design which is crucial in micro-power applications like hearing aid. The paper is organized as follows. Section II explains the beamforming algorithm. and section III gives hardware implementation details of the architecture. The proposed optimisations are discussed in section IV and the results in section V.

II. Beamforming Algorithm

The beamformer architecture implemented is based on the system proposed by Luo et al. [4], which is shown in Fig. 1. The incoming sound signal from direction θ is represented as $s(n)$. Signals $a(n)$ and $b(n)$ are sound inputs received by the front and back omnidirectional microphones respectively in the two microphone array. $\theta = 0°$ corresponds to front of the hearing aid whereas $\theta = 180°$ represents the direction directly behind the hearing aid. An adaptive logic block is used to vary the gain $G(n)$ to steer the null along different directions. Here d denotes the distance between the front and back microphones and c denotes the speed of sound. The delay T is taken as d/c. The polar pattern of $x_1(n)$ is cardioid with a null at $180°$ and that of $x_2(n)$ is cardioid with a null at $0°$. Luo et al. [4] has shown that $G(n)$ can be approximated to

$$G(n) = \frac{1 + cos\theta_{null}}{1 - cos\theta_{null}} \tag{1}$$

where θ_{null} is the direction of null. Therefore the relationship between the null and the gain is independent of the frequency of the signal. Using Least Mean Squares (LMS) (stochastic gradient descent) algorithm [4], [5] the expression of adaptive gain can be obtained as

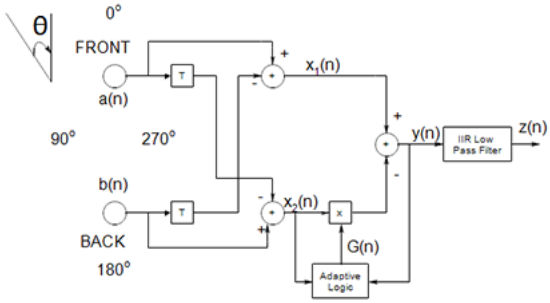

Fig. 1. Adaptive beamformer model to be implemented

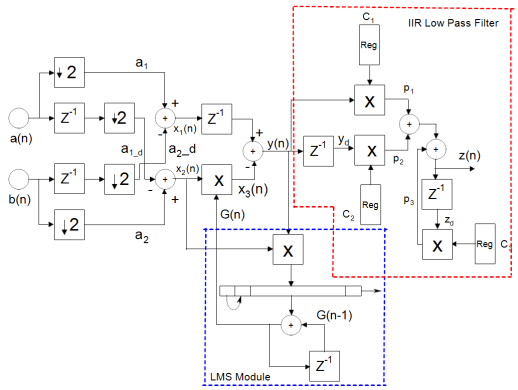

Fig. 2. Hardware realisation of the model to be implemented

$$G(n+1) = G(n) + 2\mu y(n) x_2(n) \tag{2}$$

where μ is the step parameter also known as learning rate and is implemented in the adaptive logic block. For sound coming from direction $\theta = 0°$, there is no directional cancellation of sound. Under this condition the relation between the desired sound source and the array output can be expressed as [3]

$$\left|\frac{Y(\omega,\theta)}{S(\omega)}\right| = 2\left|sin(\frac{\omega d}{c})\right| \tag{3}$$

In most of the hearing aid applications d is small when compared to the wavelength of the sound and hence $sin\theta \approx \theta$. In this case the output of directional microphone is proportional to ω which gives a magnitude response with slope of 6 dB/octave [5]. Also, the output is proportional to d, which means that reducing the separation between the microphones reduces the output level. A first order IIR low pass filter (4) is used for equalizing this high pass response from 100 Hz to 2 kHz (low frequencies).

$$H(z) = \frac{C_1 + C_2.z^{-1}}{1 - C_3.z^{-1}} \tag{4}$$

III. Hardware Description

The architecture as shown in Fig. 2 consists of a fixed beamformer module, an LMS module and a low pass equalizer filter. Fixed beamformer and LMS modules are each implemented as four state FSM and run in parallel synchronously to do adaptive beamforming. Low pass filter is implemented as an IIR filter.

$$z(n) = C_3.z(n-1) + C_1.y(n) + C_2.y(n-1) \tag{5}$$

The IIR filter was designed using MATLAB and the coefficient values used were C_1 = 0.2759, C_2 = 0.2759 and C_3 = 0.9758.

The inputs coming from the two microphones have to be delayed by $T = d/c$. The minimum sampling frequency is chosen as $F_s = c/d$ so that the incoming signal can be delayed by one sample to obtain the required delay. Since our microphones are about 1.2 cm apart and with $c = 340\ m/s$, the minimum sampling frequency is about 28 kHz. This is a very high, as typical range of voice frequencies is from 200 Hz to 4 kHz. This necessitates that the computations to run at a higher rate. Hence we downsample the signals $a(n)$ and $b(n)$ by 2 after the required delay. This helps to not only reduce the speed of computation but also reduces the dynamic power dissipation of the hardware. The microphone (ADMP401 MEMS omnidirectional microphones with OPA344 amp) output voltage ranges from 0 V to 5 V. After appropriate signal conditioning, the output of the microphone is sampled using 16-bit ADC and normalised to -1.0 to 1.0. This can be done by first removing the DC shift in the output and then by scaling the resultant. The DC shift present in the output is 2^{15} corresponding to 2.5 V. 16-bit fixed point representation with 14 bits for fractional part is chosen from the simulations in MATLAB. 16 bit ADC register is denoted as $RADC$. Let the sound values ranging between -1.0 to 1.0 be stored in register S. Then

$$S[15:0] = [\overline{RADC[15]}\ \overline{RADC[15]}\ RADC[14:1]] \tag{6}$$

If we follow the above, $2^{16} - 1$ maps to 0.99, 0 maps to -1.0 and 2^{15} maps to 0.0. Hence DC shift is removed and

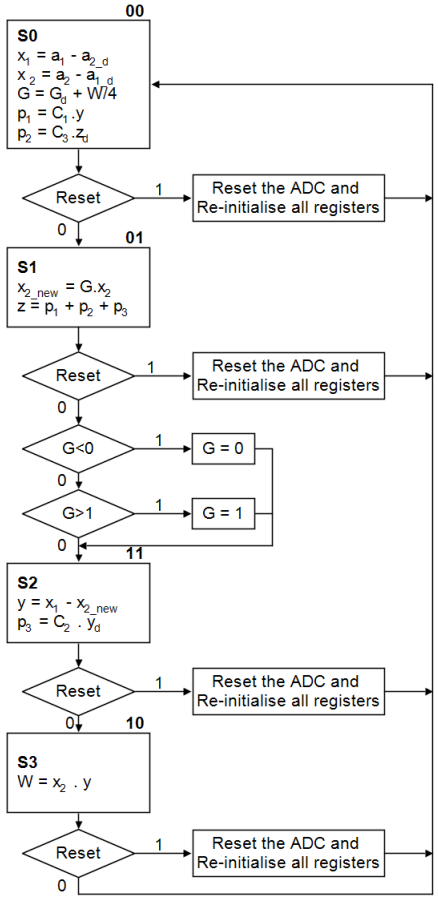

Fig. 3. ASM chart of the adaptive beamformer implemented

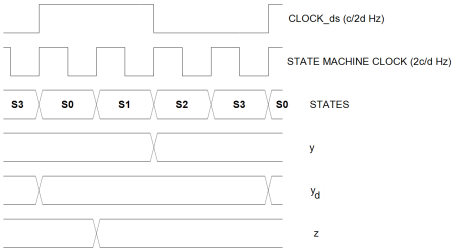

Fig. 4. Timing diagram highlighting change in y(n), y(n-1) and z(n)

STATES	OPERATIONS
S0	⊕ A1, + , x , ⊗ M2
S1	+ , A2, x , M1, + , A3
S2	+ , x
S3	x

Fig. 5. Arithmetic operations in different states of the state machine in Fig. 3. A1 and A2 are two operand 16-bit adders. M1 and M2 are radix-2 Booth's multipliers. A3 is a three operand 32-bit adder

the sound samples now fall in -1 to 0.99. There is a loss of accuracy by one bit ($RADC[0]$) which is negligible but makes the hardware efficient.

The algorithmic state machine (ASM) chart of the hardware is shown in Fig. 3. This ASM chart describes the beamformer operation after the downsampling is performed. All the four states of the state diagram must be executed before the arrival of the new sample from the ADC and hence the state machine in Fig. 3 is running at $2c/d$ (4 times $c/2d$). The inputs to the LMS module are two 16-bit numbers, $x_2(n)$ and $y(n)$, which are multiplied and the product is then shifted to the right by 2 bits, to multiply with $2\mu = 0.25$ and is added to the previous value of gain, G_d to get $G(n)$ (2). The value of G is restricted between 0 to 1. The value of final output, $z(n)$ is obtained from the low pass filter. The timing diagram is shown in Fig. 4. New value of $y(n)$ is obtained in state $S2$ whereas new value of $z(n)$ is obtained in $S1$ in the next iteration of the ASM. Initially, Radix-2 Booth's multiplier was used for signed multiplication. It takes 32 cycles to multiply two 16 bit fixed point numbers. The 32-bit product is truncated to 16-bits while updating the value of G. This dataflow results in 5 multiplications, three 16-bit addition/subtraction of two operands and one 3 operand 32-bit addition for computation of $z(n)$. From the careful scheduling of operations, we can dedicate 2 multipliers, M1 active in all 4 states and M2 only in $S0$ as shown in Fig. 5.

IV. Proposed Optimisations

Two of coefficients of designed IIR low pass filter, C_1 and C_2 are equal as seen in section III. This feature is used to reduce the hardware. The ASM representation of the modified architecture is shown in Fig. 6. Here, the sum of $y(n)$ and

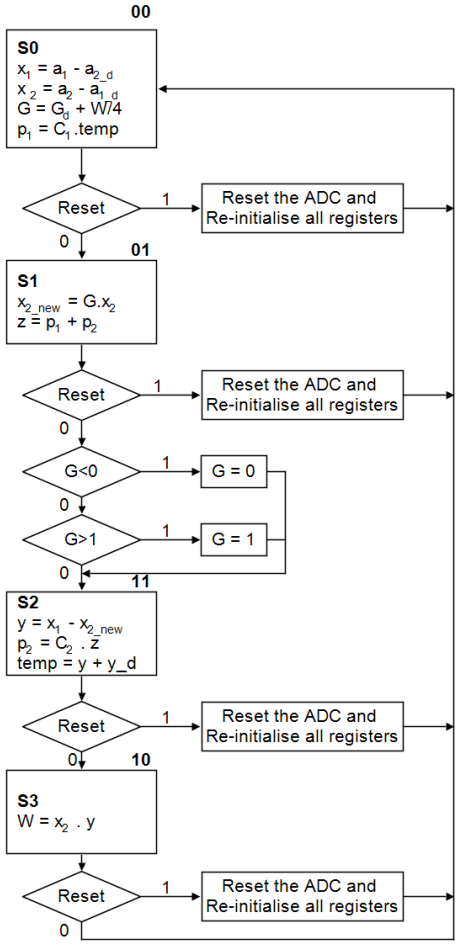

Fig. 6. ASM of the optimised hardware

STATES	OPERATIONS
S0	+ , + , x
S1	A1, A2, x , M1, + , A3
S2	+ , + , x
S3	x

Fig. 7. Arithmetic operations in different states for the optimised ASM

$y(n-1)$ is stored as a variable $temp$, which is then multiplied by the common coefficient C_1. From Fig. 4, we see that in $S2$ both $y(n)$ and $y(n-1)$ are available, hence in this state $temp$ is computed. Multiplication of $temp$ with C_1 is done in state $S0$ and the multiplication of $z(n)$ and C_2 is shifted to $S2$. $z(n)$ is computed in $S1$ and then in next state $S2$, this value is multiplied with C_2, which serves as $C_2.z(n-1)$ when the $z(n)$ is computed in $S1$ of next iteration of the ASM. Also, $C_1.temp$ computed in state $S0$, serves as $C_1.y(n)+C_1.y(n-1)$ for computation of $z(n)$ in next state $S1$. From Fig. 7, we see that there is only one multiplication operation in every state. Hence only one multiplier is needed which is active in all the states.

Radix-2 Booth's multiplier was replaced by a radix-4 Booth's multiplier. It takes half the number of cycles when compared with radix-2 multiplier and hence needs 16 clock cycles and reduces the dynamic power consumption without much change in the hardware resources. Faster carry save adders were used for addition. Gray coding was used to encode the states to reduce the glitches.

V. Implementation Results

The beamforms with nulls at 90°, 120° and 180° were obtained from the system implemented and are shown in Fig. 8. The radius of polar plots in Fig. 8 represents the ratio of output power to input power. Patterns shown do match the expected patterns.

The architecture was first implemented on Xilinx Artix 7 FPGA (Xilinx Nexys 4 FPGA board with xc7a100tcg324-1). An experiment to check the functioning of the algorithm was conducted, in which the desired speech signal was arriving from a source at 0° azimuth, while noise was arriving from different directions given in Table I. Two kinds of noise were used to test the algorithm, a 1.8 kHz monotone and another speech signal. The Table I also shows the SNR improvement (SNRe) values obtained. It can be seen that the output coming from the beamformer has significant reduction in the noise.

After completing the functional verification of the architecture on Xilinx Artix 7 FPGA it was implemented in Cadence Encounter using $0.18\ \mu m$ SCL libraries [6]. The layout implemented was analysed in terms of area reports obtained from Cadence Encounter and power reports obtained from Cadence Voltus.

1) Area Reports: The area report for each of the optimizations is given in Tabe II. It can be seen from the Table II that, as the number of multipliers reduced from 5 to 2 and then to 1, the area and number of standard cells required reduced appreciably. It can be noticed that there is not much change in the area on moving from radix-2 to radix-4 Booth's multiplier.

2) Power Reports: The power report is summarized in Table III. The total power reduces as the number of multipliers reduced from 5 to 2 and then to 1. Further reduction happens on going from radix-2 to radix-4 Booth's multiplier. As the number of multipliers reduces, static power reduces. The dynamic power also reduces considerably as the switching activity of the hardware reduces with reduction in clock frequency.

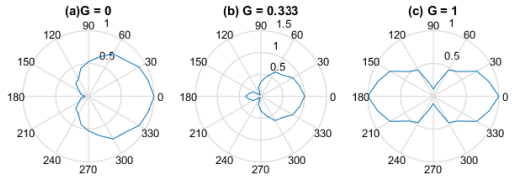

Fig. 8. Beamform patterns obtained from the implementation

TABLE I
Improvements in SNR Values

Type of Noise Signal	Direction of Incoming Noise (Degrees)	SNRe (dB)
Monotone Signal	180	20.81
	150	22.10
	105	21.21
Speech Signal	180	15.08
	150	15.12
	105	14.92

TABLE II
Area Comparison

Implementation	Area (μm^2)	Cell number
5 Multipliers-Radix2	102879.682	3290
2 multipliers- Radix2	69895.902	2146
1 Multiplier Radix2	57191.961	1766
1 Multiplier Radix-4	54216.081	1636

TABLE III
Power Comparison

Implementation	Static power (μW)	Dynamic Power (μW)	Total Power (μW)
5 multipliers Radix-2	1.39	839.70	841.10
2 multipliers radix-2	0.92	191.32	192.24
1 multiplier radix-2	0.72	106.62	107.34
1 multiplier radix-4	0.68	59.86	60.54

VI. Conclusion

An area and power optimized ASIC implementation of adaptive beamforming technique was done and the improvements were highlighted in this paper. The hardware was initially prototyped using FPGA and later implemented with SCL $0.18\ \mu m$ Technology libraries. The functionality of the algorithm was verified by plotting beamform patterns for different noise directions and the SNR improvements were analyzed. The algorithm performed as expected and total area and power consumption of the design reduced considerably with the proposed hardware optimisations.

References

[1] J. G. Desloge, W. M. Rabinowitz, and P. M. Zurek, "Microphone-array hearing aids with binaural output. i. fixed-processing systems," *IEEE Transactions on Speech and Audio Processing*, vol. 5, no. 6, pp. 529–542, 1997.

[2] W. Soede, A. J. Berkhout, and F. A. Bilsen, "Development of a directional hearing instrument based on array technology," *The Journal of the Acoustical Society of America*, vol. 94, no. 2, pp. 785–798, 1993.

[3] G. W. Elko and A.-T. N. Pong, "A simple adaptive first-order differential microphone," in *Applications of Signal Processing to Audio and Acoustics, 1995., IEEE ASSP Workshop on*. IEEE, 1995, pp. 169–172.

[4] F.-L. Luo, J. Yang, C. Pavlovic, and A. Nehorai, "Adaptive null-forming scheme in digital hearing aids," *IEEE Transactions on signal processing*, vol. 50, no. 7, pp. 1583–1590, 2002.

[5] J. M. Kates, *Digital hearing aids*. Plural publishing, 2008.

[6] "Semi-conductor laboratory," http://www.scl.gov.in/.