Università degli Studi di Pavia

FACOLTÀ DI INGEGNERIA
Corso di Scuola di Dottorato in Microelettronica

Tesi di Dottorato

ANALOG VOICE ACTIVITY DETECTION

Candidato:
Marco Croce

Relatore:
Prof. Piero Malcovati

Correlatori:
Prof. Andrea Baschirotto
Dr. Lorenzo Crespi

Anno Accademico 2015–2018

Contents

List of Figures **iii**

List of Tables **vii**

Abstract **ix**

Introduction **2**

1 VAD Techniques **3**

1.1 Zero Crossing Methods 3

1.2 Energy-Based Methods 4

1.3 Linear Prediction Methods 5

1.4 Single-Frequency Filtering Methods 6

1.5 Neural Network Methods 8

2 Microphone Preamplifiers **13**

2.1 Switched-Resistor Preamplifier 15

2.2 Off-Transistor Preamplifier 16

2.3 Measurement Results 17

3 Analog VAD Circuit **21**

3.1 Programmable-Gain Amplifier 24

3.2 Signal Energy Computation 29

3.3 Energy Averaging and Noise Level Computation 38

4 Measurement Results **47**

Conclusions **57**

Bibliography **59**

List of Figures

1.1 VAD based on the coherence function 6
1.2 Multi-resolution stacking VAD [1] 9
1.3 Recurrent neural network VAD [2] 10
1.4 Modulation frequency feature extraction VAD [3] 11
1.5 Feed-forward neural network VAD [4] 12

2.1 Block diagram of a capless preamplifier 14
2.2 Switched-resistor preamplifier 14
2.3 Off-transistor preamplifier 16
2.4 Microphotograph of the test chip containing both preamplifiers 17
2.5 Frequency response of both preamplifiers in all of the gain configurations . 18
2.6 *THD* of both preamplifiers in all of the gain configurations 19
2.7 Spectrum of both preamplifiers for a 1-kHz,-1 $\mathrm{dB_{FS}}$ input signal in 0-dB gain configuration 19
2.8 Input-referred noise of both preamplifiers in all of the gain configurations . 20
2.9 Simulated variations over PVT of the off-transistor preamplifier frequency response in 0-dB gain configuration 20

3.1 Signal processing chain in the absence of voice 21
3.2 Signal processing chain in the presence of voice 22
3.3 Typical audio input signal (a) and zoom over a frame of 16 ms (b) 22
3.4 Block diagram of the analog VAD circuit 24
3.5 Schematic of the Miller opamp 25
3.6 Feedback network around the PGA 26
3.7 Poly-resistor model 26

3.8 Switched-resistor implementation (a) and clock signals (b) 27
3.9 Simulated pen-loop gain and phase of the PGA with no load 28
3.10 Simulated open-loop gain and phase of the PGA with load 28
3.11 Circuit used to verify the PGA stability in transient conditions 29
3.12 Transient simulation to verify stability with common-mode current pulses applied at the output of the PGA 30
3.13 Transient simulation to verify stability with differential current pulses applied at the output of the PGA 31
3.14 Basic idea for implementing the square operation [5] 32
3.15 Adopted circuit for implementing the square operation 33
3.16 Squarer output current and ideal square waveform overlapped (top) obtained with a ramp input signal (bottom) implemented with a DC sweep. 34
3.17 Resettable integrator 35
3.18 Integrator output offset 36
3.19 Integrator open-loop gain and phase with 40-pF load 36
3.20 Biasing circuit for the PGA, the squarer, and the integrator 37
3.21 Overall schematic of PGA, squarer, and integrator 37
3.22 Schematic of the circuit used for VAD generation 38
3.23 NL update circuit during the integration period (a) and during the update period with C_β_1 (b) 39
3.24 Energy averaging circuit during the integration period (a) and during VAD decision (b) 41
3.25 Clock phases: integrator capacitance reset (a), evaluation phase (b), selection between SW_comp11 and SW_comp12 (c), comparator clock (d), VAD decision (e) 43
3.26 Circuit configuration during the integration period (a) and during the first comparison (b) 44
3.27 Circuit configuration during the second comparison 45
3.28 Dynamic-latch comparator 45
3.29 Comparator offset over 100 Montecarlo simulations 46

4.1 Microphotograph of the test chip 48
4.2 Test chip layout detail: CT resistor (above) and SW resistor (bottom) . . . 49

4.3 Measured clock phases 50
4.4 Audio frame used for testing 51
4.5 Errors in the VAD signal with respect to the ideal model for the CT-resistor circuit (a) and the SW-resistor circuit (b) 53

List of Tables

2.1 Preamplifier gain and pole frequency programmability 15

3.1 Gain programmability of the PGA (conventional resistor) 26
3.2 Gain programmability of the PGA (switched resistor) 27
3.3 Possible values of C_β_1 and C_β_2 40
3.4 Possible values of C_{ratio2} 42

4.1 Measured VAD errors with Gain = 12 dB, Integration Period = 16 ms, Integration Capacitance = 40 pF, $\beta_1 = 0.95$, $\beta_2 = 0.995$, and $th_{SP} = 0.42$. 52
4.2 Comparison with the state-of-the-art 54
4.3 Measured VAD errors with Gain = 12 dB, Integration Period = 16 ms, Integration Capacitance = 40 pF, $\beta_1 = 0.99$, $\beta_2 = 0.95$, and $th_{SP} = 0.69$ . . 55
4.4 Measured VAD errors with Gain = 6 dB, Integration Period = 16 ms, Integration Capacitance = 20 pF, $\beta_1 = 0.99$, $\beta_2 = 0.95$, and $th_{SP} = 0.57$ . . 55
4.5 Measured VAD errors with Gain = 0 dB, Integration Period = 16 ms, Integration Capacitance = 10 pF, $\beta_1 = 0.99$, $\beta_2 = 0.95$, and $th_{SP} = 0.82$ for the CT-resistor circuit and $th_{SP} = 0.85$ for the SW-resistor circuit 55
4.6 Measured results 57

Abstract

This Thesis presents a Voice Activity Detection (VAD) system, entirely implemented in the analog domain with a 180-nm CMOS technology. The circuit features a current consumption of 0.9 μA from a 1.8-V supply voltage. The VAD system is composed of three main blocks: a preamplifier, a signal energy computation block, and a VAD decision block. The audio signal coming from the microphone is amplified and filtered by a preamplifier that features a variable gain ranging from -12 dB to $+12$ dB with 6-dB steps and a bandpass transfer function with poles at 300 Hz and 6.8 kHz. The preamplifier has been implemented both with continuous-time resistors to allow large decoupling capacitors at the input, where the gain is set by the resistance ratio, and with switched resistors to reduce the chip area, where the gain is set by capacitance ratio. The second block of the circuit computes the audio signal energy in the analog domain, exploiting the transistor quadratic current-voltage relation to square the signal and integrating the resulting current with a resettable capacitance. The final block produces the VAD signal. In this block the computed signal energy is used for two different purposes: determine the background noise level and the energy average. The noise level is constantly updated and compared with the averaged energy to provide the VAD signal. The measurement results on an integrated prototype demonstrate that the analog VAD can achieve performances comparable with state-of-the-art digital implementations, but with much lower power consumption.

Introduction

Voice Activity Detection (VAD), also known as Speech Activity Detection or Speech Detection, is the identification of the presence or absence of human speech in an audio signal. Generally, VAD is a pre-processing wake-up mechanism used to enable and disable the signal processing interface at appropriate times, avoiding unnecessary coding or transmission of silence packets and saving on computation and network bandwidth. This mechanism can be adopted for different purposes, such as automatic speech recognition, speaker verification, speech enhancement, voice operation switch, voice over internet protocol, etc. [6–9]. The main challenges in these applications, where the audio acquisition channel has to be always on, are the power consumption, the poor signal-to-noise ratio (SNR), and the wide amplitude variation of speech and non-speech signals. There are many techniques to detect human voice. Most of them consist of feature extraction and classification algorithms, that can be implemented either in the analog or in the digital domain. The analysis of the audio signal features provides an indication on speech presence or absence. The algorithms commonly used can be based on time-based and frequency-based approaches [5]. Typically, the audio signal is processed over non-overlapping frames with a duration between 10 ms and 20 ms. A VAD algorithm should feature the following characteristics:

- Low power consumption: as an always-on and real-time application, the VAD algorithm complexity and power consumption must be low;
- Adaptability: the ability to handle non-stationary background noise variations improves robustness;
- Decision rule: a physical property of the incoming audio signal frames is used to detect the presence or absence of speech and give consistent and accurate judgement in classification.

This Thesis, which describes a VAD system entirely implemented in the analog domain, is organized as follows. Chapter 1 covers in a general way and with just introductory purposes

some of the most commonly used VAD algorithms available in the literature, highlighting the decision rule and the final purpose of each solution. Chapter 2, then, describes a preliminary study of the microphone preamplifier, which represents the first circuit block in any VAD system. Two solutions to implement capless (without external decoupling capacitor) preamplifiers, based on off-transistors and switched resistors, respectively, have been designed, integrated, and tested. The proposed analog VAD circuit is then described in detail in Chapter 3. All of the blocks required for implementing the VAD system are analyzed, reporting all of the design details and the simulation results. Finally, Chapter 4 reports all of the experimental results achieved from the prototype integrated in a 180-nm CMOS technology. The obtained performances are compared with the state-of-the-art [10–14].

Chapter 1

VAD Techniques

Voice Activity Detection (VAD) algorithms usually have as input the audio stream to be processed and as output a digital signal, which flags the presence (1) or absence (0) of voice. The first processing step required for VAD is the extraction and evaluation of signal features, which identify the presence of voice over background noise or other interfering signals. However, for practical applications the feature selection has to take in account limited hardware and latency constraints, which means limited power, area, and time to get the desired information from an audio frame. Once some distinctive features have been identified, in the second and last processing step, a detection rule is applied to the extracted feature, leading to the final VAD signal. In the literature there are several techniques that have been developed for implementing VAD, which can be classified in the following categories:

- Zero crossing methods;
- Energy-based methods;
- Linear prediction methods;
- Single-frequency filtering methods
- Neural network methods.

A general description of these methods will be provided in the following of this chapter.

1.1 Zero Crossing Methods

The Zero Crossing Rate (ZCR) method [15] is based on the detection of number of changes of sign in the audio signal amplitude during the analyzed frame. For each frame n,

it is possible to define the function $Z(n)$ as:

$$Z(n) = \sum_{m=1}^{M} \frac{|\operatorname{sign}[x(m)] - \operatorname{sign}[x(m-1)]|}{2} \tag{1.1}$$

where $x(\cdot)$ is the audio signal, M is the number of samples per frame and $\operatorname{sign}(\cdot)$ is the sign function, defined as follow:

$$\operatorname{sign}(x) = \begin{cases} 1 & for\ x \geq 0; \\ -1 & for\ x < 0. \end{cases} \tag{1.2}$$

The ZCR method uses the frequency features of the signal to build the VAD decision rule, based on the assumption that voice components are located at low frequencies and noise components at high frequencies. If the number of zero crossings $Z(n)$ is low, the segment analyzed is classified as voice, whereas if it is high it is classified as noise.

1.2 Energy-Based Methods

The most popular and widely used techniques in speech detection are energy-based [16–18], since they are easy to implement and requires low computational complexity. Generally, the energy of voiced speech segments is higher compared to unvoiced segments and, as for ZCR methods, voiced speech has most of its energy in the lower frequencies. There are different ways to represent the energy of a signal:

$$E(n) = \sum_{k=1}^{K} \log[x^2(k,n)], \quad E(n) = \sum_{k=1}^{K} x^2(k,n), \quad E(n) = \sum_{k=1}^{K} |x^2(k,n)|, \tag{1.3}$$

where K is the total number of samples per frame, n is the current frame, and $x(\cdot)$ is the audio signal. The first equation on the left represents the logarithmic short-term energy, the second equation represents the squared short-term energy, and the third equation represents the absolute short-term energy. All of these are full-band energy examples. Sometimes windowing can be applied to attenuate unwanted frequency components [19]:

$$E(n) = \sum_{k=-\infty}^{\infty} x(k) \cdot h(n-k) \tag{1.4}$$

$$h(n) = \begin{cases} 0.54 - 0.46 \cdot \cos\left(\dfrac{2\pi n}{N-1}\right) & 0 \leq n \leq N-1 \\ 0 & \text{otherwise} \end{cases} \tag{1.5}$$

where, for example, the hamming window $h(\cdot)$ is used. The calculated energy of the incoming signal is compared with a threshold value, that can be fixed or adaptable to background noise variations, to decide if the incoming frame is voiced or unvoiced. An example where the threshold value is updated continuously can be found in [18]:

$$E_{th\_new} = E_{th\_old} \cdot (1 - \alpha) + E_{new} \cdot \alpha; \tag{1.6}$$

where E_{th_new} is the updated threshold value, E_{th_old} is the previous threshold value, E_{new} is the energy of the current frame and α has a value between 0 and 1.

1.3 Linear Prediction Methods

Linear prediction (LP) algorithms [20–22] are generally used for speech recognition, speaker recognition, speech coding, speech synthesis, and speaker verification. As suggested by the name, this technique aims to estimate the current signal value $x(n)$ with a linear combination of past events:

$$\tilde{x}(n) = \sum_{k=1}^{K} \beta_k \cdot x(n-k) \tag{1.7}$$

where K is the total number of samples per frame, n is the current frame, and β_k are the predictor coefficients, obtained minimizing the mean square error of the prediction error:

$$e(n) = x(n) - \tilde{x}(n). \tag{1.8}$$

LP methods have been extensively used in speech detection, for example with the well known coherence function defined as:

$$C_{xe}(n, f) = \frac{P_{xe}(n, f)}{\sqrt{P_{xe}(n, f) \cdot P_{ee}(n, f)}} \tag{1.9}$$

where $P_{ee}(n, f)$ is the spectral density of $e(n)$ and $P_{xe}(n, f)$ is the inter-signal spectral density between $x(n)$ and $e(n)$.

Figure 1.1 shows the block diagram of a VAD system where the averaged coherence function is used. The coherence function output amplitude can have a value close to one in case of a purely noise signal, close to zero if the signal contains speech, and a middle value for unvoiced signals. Finally, a thresholding decision rule is implemented:

$$\begin{aligned} |C_{xe}(n, f)| \geq th \quad &\Rightarrow noise \\ |C_{xe}(n, f)| < th \quad &\Rightarrow speech \end{aligned} \tag{1.10}$$

where th is the threshold adopted.

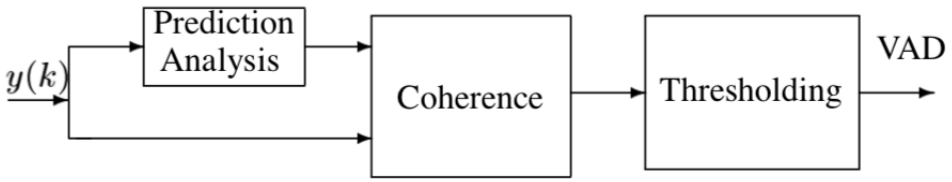


Figure 1.1: VAD based on the coherence function

1.4 Single-Frequency Filtering Methods

The single frequency filtering (SFF) [23] method is based on the assumption that noise energy is equally distributed over frequency, while speech energy has a non-uniform distribution. Therefore, the SNR of the speech signal is higher at certain frequencies compared to other frequency regions. Defining $S(f)$ as the signal amplitude and $N(f)$ as the noise amplitude as a function of frequency, the SNR can be computed as follows:

$$SNR_a = \int_{f_0}^{f_L} \frac{S^2(f)}{N^2(f)} df, \quad SNR_b = \sum_{i=0}^{L-1} \frac{\int_{f_i}^{f_{i+1}} S^2(f) df}{\int_{f_i}^{f_{i+1}} N^2(f) df}, \\ SNR_c = \frac{\int_{f_0}^{f_L} S^2(f) df}{\int_{f_o}^{f_L} N^2(f) df}, \tag{1.11}$$

where (f_i, f_{i-1}) are L non overlapping frequency intervals. The following inequality holds:

$$SNR_a \geq SNR_b \geq SNR_c. \tag{1.12}$$

In [24] a VAD system based on SFF is presented, where the input signal is sampled at frequency f_s and its differenced discrete time version $x(n) = s(n) - s(n-1)$ is multiplied by a complex sinusoid:

$$x_k(n) = x(n) \cdot e^{j\overline{w}_k n}, \tag{1.13}$$

in which $\overline{w}_k = 2\pi \overline{f}_k / f_s$ is the normalized frequency. After this multiplication the differenced signal is processed with a single-pole filter with the following transfer function:

$$H(z) = \frac{1}{1 + rz^{-1}}. \tag{1.14}$$

The pole is located on the real negative axis at a distance from the origin equal to half of the sampling frequency. The magnitude of the signal $y_k(n)$ coming from the filter is given by:

$$m_k(n) = \sqrt{y_{kr}^2(n) + y_{ki}^2(n)}, \tag{1.15}$$

where y_{kr} is the real part and y_{ki} is the imaginary part of $y_k(n) = -ry_k(n-1) + x_k(n)$ and $m_k(n)$ can be seen as the magnitude of the signal $x_k(n)$ filtered at the desired frequency $f_k = f_s/2 - \overline{f}_k$.

The distribution of the noise power across frequencies generally is uniform, but, for non-stationary noise, the power is not uniformly distributed. Weighting the signal at each frequency with the floor value is a way to compensate the non-stationary noise effect. Assuming that in the first part of each signal frame speech is not present, the mean of the magnitude of this portion is used to compute the normalized weight w_k at each frequency f_k:

$$w_k = \frac{\frac{1}{\mu_k}}{\sum_{p=1}^{N} \frac{1}{\mu_p}}, \tag{1.16}$$

where N is the total number of channels. The compensation is achieved multiplying the magnitude $m_k(n)$ with the weighting coefficient w_k at each frequency. The energy of the signal at each instance can be approximated through the mean $\mu(n)$ of the square of the weighted component magnitudes across frequency, which is higher for speech than for noise when a speech signal is present. Another quantity that exhibits the same behavior is the standard deviation $\sigma(n)$ of the square of the weighted component envelopes computed across frequency. To highlight the contrast between speech and non-speech regions the following combination of $\mu(n)$ and $\sigma(n)$ can be used:

$$\delta(n) = \sqrt[M]{|\sigma^2(n) - \mu^2(n)|}. \tag{1.17}$$

The decision on speech presence or absence is based on the comparison between a threshold $\theta(n)$ and temporally smoothed $\delta(n)$ values. The threshold is defined as follows:

$$\theta = \mu_\theta + 3\sigma_\theta, \tag{1.18}$$

and is updated on every utterance to keep track of background noise changes. To determine the smoothing window size l_w, the dynamic range ρ based on the signal energy is computed on each frame m of 300 ms length:

$$\rho = 10 \cdot log_{10}\left[\frac{\max_m(E_m)}{\min_m(E_m)}\right] \quad \Rightarrow \quad \begin{cases} l_w = 400\,ms & \rho < 30, \\ l_w = 300\,ms & 300 \leq \rho \leq 40, \\ l_w = 200\,ms & \rho > 40. \end{cases} \tag{1.19}$$

Once the window size is obtained, the averaged value of $\delta(n)$ can be determined and compared with the threshold:

$$\begin{aligned} d(n) = 1, \quad for \quad \overline{\delta}(n) > \theta \quad \text{(presence of speech)} \\ d(n) = 0, \quad for \quad \overline{\delta}(n) \leq \theta \quad \text{(absence of speech)} \end{aligned} \tag{1.20}$$

This decision algorithm can be smoothed counting the number of times per frame in which $d(n)$ is equal to one. Calling $d_f(n)$ the percentage of ones in the window, if $d_f(n)$ is higher than a threshold η, the frame contains speech, otherwise it is classified as a non-speech frame. Making a comparison between this method and adaptive multi-rate (AMR) methods, the length of the frames processed has to be lowered to 10 ms.

1.5 Neural Network Methods

Neural networks in general can be defined as structures built to emulate human brain activity. The input signal fed to the network goes through different layers that simulate neural connections. These sophisticated predictive models are composed of an input layer, a certain number of hidden layers, depending on the complexity and precision of the prediction, and an output layer which provides the desired result. An important and desirable characteristic of neural networks for VAD is the ability to classify unstructured data based on its features in the frequency or time domain. Widely used in neural networks is the sigmoid function:

$$S(x) = \frac{1}{1 + e^{-x}} \tag{1.21}$$

due to its shape and properties.

In [1] Neural Networks are used as base classifier of a multi-resolution stacking (MRS) learning framework, whose operation principle is highlighted in Figure 1.2. As can be seen, the training process of a building block is based on the previous building block predictions $\hat{y}$ and on the input acoustic feature x_m. In this way, even if the number of building blocks is increased, the informations carried by the original features will not be lost and $\hat{y}$ will add information improving performance. The hard decision $\bar{y}_m$ is the final prediction:

$$\bar{y}_m = \begin{cases} 1 & \hat{y}_{S,1,m} \geq \delta, \\ 0 & otherwise \end{cases} \tag{1.22}$$

where $\hat{y}_{S,1,m}$ is the output of a MRS of S building blocks related to the input x_m and δ is a decision threshold which can assume values from 0 to 1. The dynamic neural network used

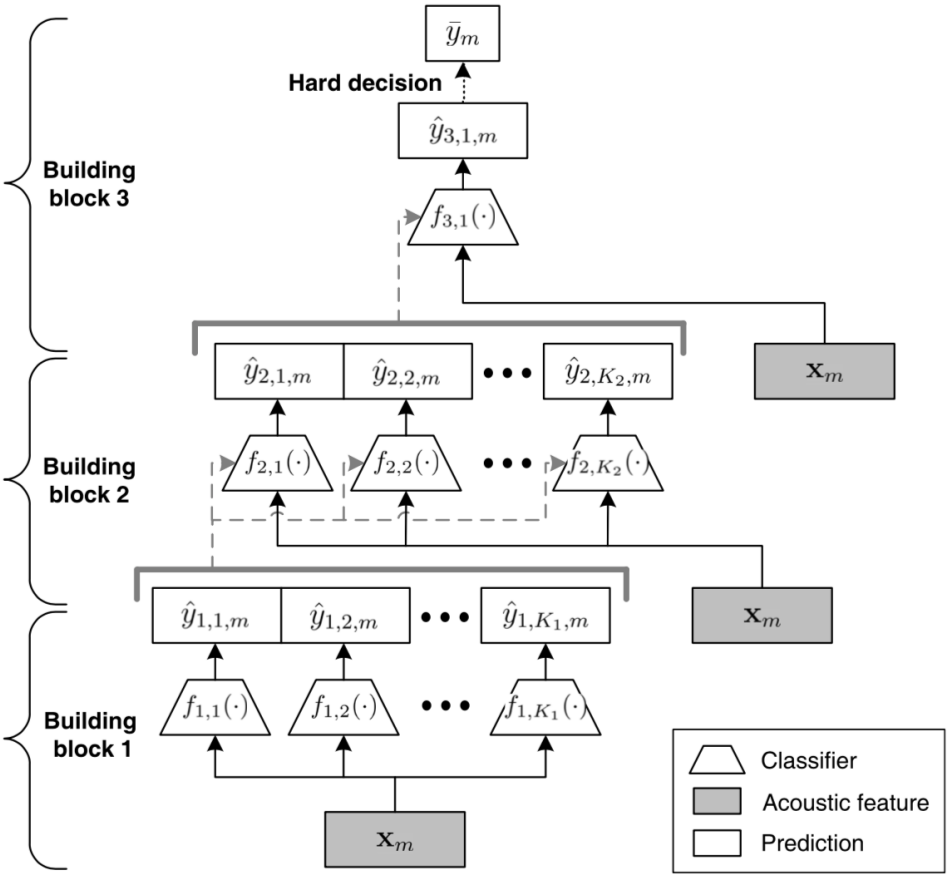


Figure 1.2: Multi-resolution stacking VAD [1]

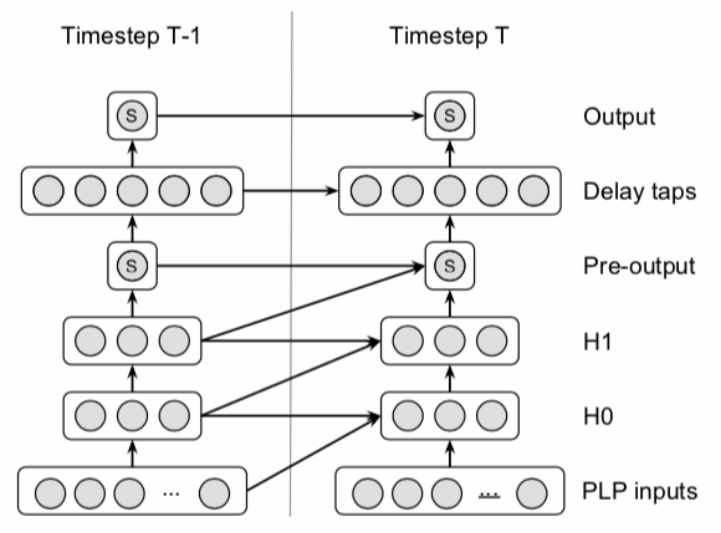


Figure 1.3: Recurrent neural network VAD [2]

in this paper can be described as follows:

$$y = h_0\Big(h_{(L)}\Big(...h_{(l)}\Big(h_{(1)}\Big(x^{(0)}\Big)\Big)\Big)\Big); \tag{1.23}$$

where L denotes the number of hidden layers, $h_0(\cdot)$ is the final layer, $h_{(l)}(\cdot)$ denotes a group of non-linear mapping functions and $x^{(0)}$ represents the input features. A rectified linear function is used for the hidden layers ($y = max(0, x)$) to better handle local patterns, while the sigmoid function is used for the output layer.

A recurrent neural network is presented in [2], where each node combines the inputs with a quadratic polynomial function:

$$v(x) = f(x^T \cdot W_Q \cdot x + w_L^T \cdot x + w_B); \tag{1.24}$$

where $v(x)$ is the output node, W_Q and x_L represent the sparse triangular upper matrix containing the quadratic weights and the vector containing the linear weights, respectively, while w_B is a bias. As shown in Figure 1.3 the input of each node at time-step T is a combination of the previous node output in the same time-step T and in time-step $T-1$, which will be more useful in the evaluation due to their high correlation and to provide temporal continuity. *Output* and *Pre-output* use a hyperbolic tangent non-linearity instead of the quadratic polynomial function.

In [3] the input signal is processed as shown in Figure 1.4 to obtain its modulation

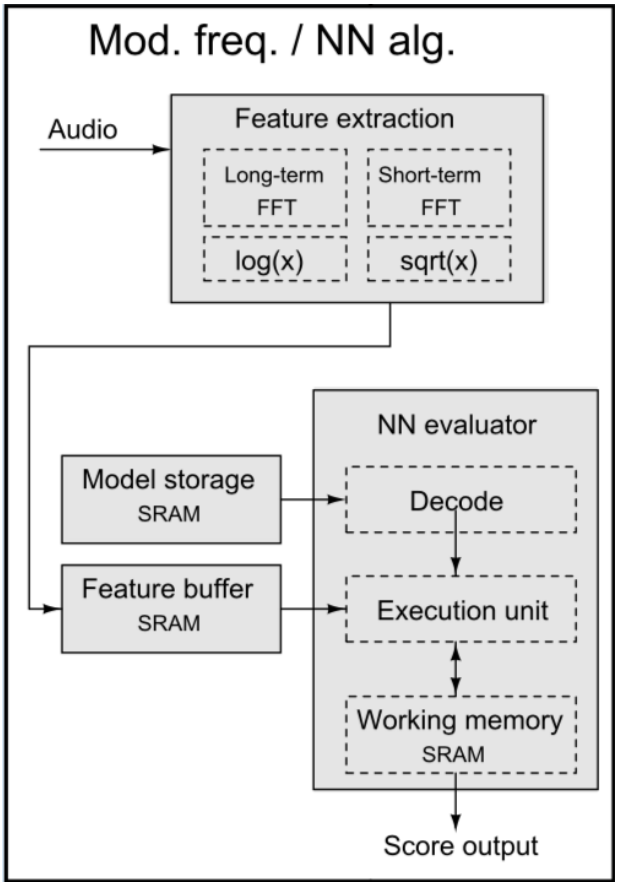


Figure 1.4: Modulation frequency feature extraction VAD [3]

frequency features, which will be sent to the neural network (NN) evaluator where the difference between the last two layers is used to determine the presence or absence of speech.

Figure 1.5 shows the neural network implemented in [4]. In this paper the nodes of the hidden layer and output layer use the sigmoid activation function and all the layers are connected through weights. The output of the j-th hidden layer node related to the p-th input pattern can be expressed as follow:

$$h_{p,j} = S\Big(b_j + \sum_{i=0}^{n} w_{j,i} \cdot x_{p,i}\Big); \tag{1.25}$$

where $S(\cdot)$ is the sigmoid function, $w_{j,i}$ are the weights related to the j-th hidden layer and the i-th input node and b_j is the bias of the related neuron. From this we can evaluate the output of the neural network:

$$y_{p,k} = S\Big(b_k + \sum_{j=0}^{m} w_{k,j} \cdot x_{p,j}\Big); \tag{1.26}$$

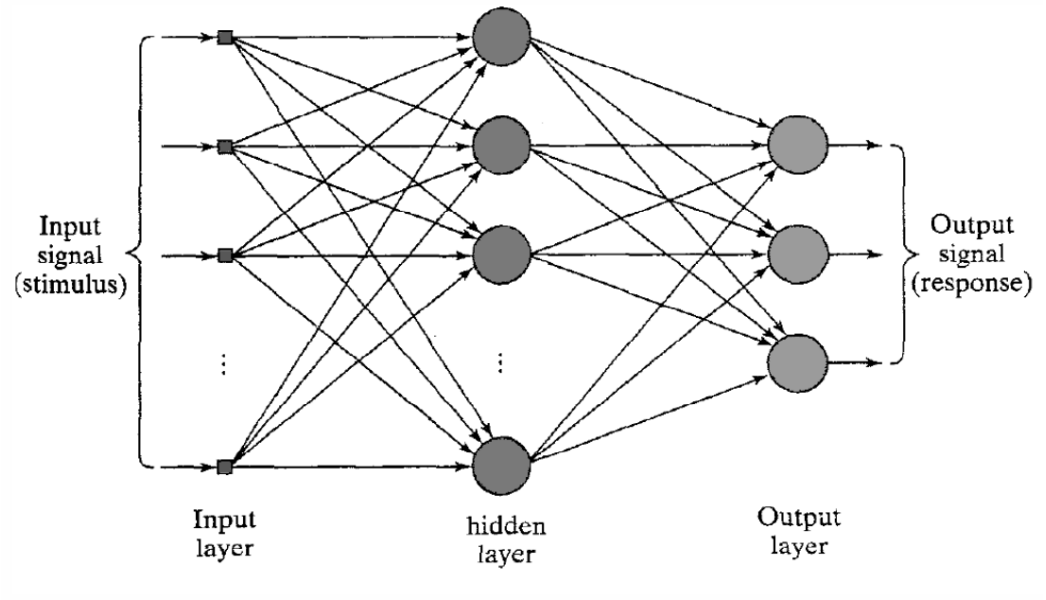


Figure 1.5: Feed-forward neural network VAD [4]

where $w_{k,j}$ are the weights related to the j-th hidden node and k-th output node and b_k is the bias of the k-th output node. The output target vector has three different configurations:

- [1 0 0] voiced input signal;
- [0 1 0] unvoiced input signal;
- [0 0 1] silence.

Chapter 2

Microphone Preamplifiers

In order to implement any of the VAD algorithms described in Chapter 1, as the very first step, it is necessary to read-out the audio signal detected by a microphone. Therefore, the implementation of microphone preamplifiers is a fundamental prerequisite for realizing a VAD system. The different features of microphones, such as common-mode (CM) voltage, single ended or differential output signal, and output signal amplitude, should not affect the preamplifier and, hence, the VAD system performance. Common-mode voltage differences can be overcome with a low-frequency *ac* coupling between the microphone and the preamplifier, generally implemented with a bulky and expensive external capacitor. A cheaper solution could be the implementation of a so called *capless* (no external capacitor) preamplifier [25, 26]. Flat frequency response is required in the audio band from 20 Hz to 20 kHz, leading to a low-frequency high-pass pole lower than 1 Hz. The variability of the microphone output signal amplitude can be tackled with a programmable gain in the amplifier, to provide the optimal signal amplitude to the following blocks in the signal processing chain.

The block diagram of a single-ended inverting preamplifier is shown in Figure 2.1. In this implementation the gain is set by the ratio C_I/C_F, while the low frequency pole is implemented by the feedback capacitance C_F and resistance R_F. To achieve a pole frequency lower than 1 Hz, even with a capacitance of 80 pF, R_F has to be in the range of hundreds of MΩ, a prohibitive value for conventional integrated resistors. Two solutions have, therefore, been investigated to achieve such high resistance value and maintain a *dc* feedback path in the preamplifier: a switched resistor and a transistor in the off state.

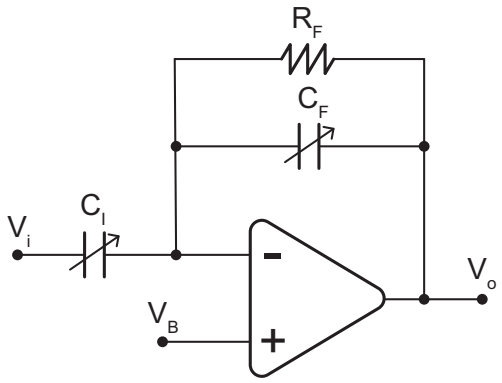


Figure 2.1: Block diagram of a capless preamplifier

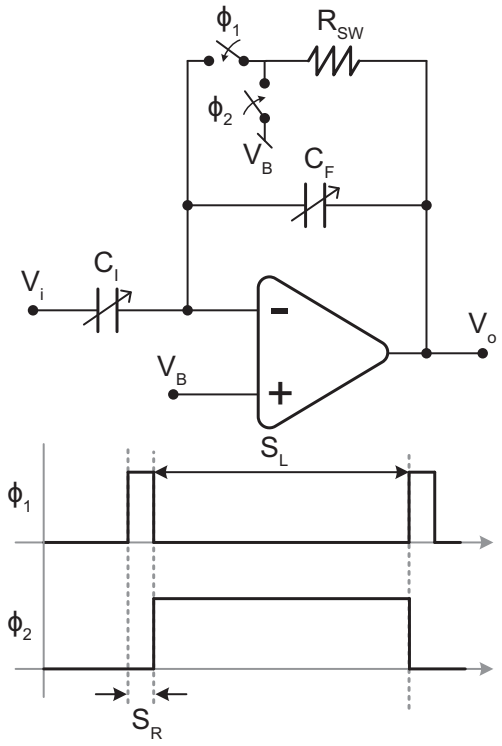


Figure 2.2: Switched-resistor preamplifier

Table 2.1: Preamplifier gain and pole frequency programmability

f_{HP} [Hz] @ MF				C_I [pF]	C_F [pF]	G [dB]
100	**200**	**300**	**400**			
4	2	1.3	1	40	80	−6
8	**4**	2.7	2	40	40	0
8	**4**	2.7	2	80	40	6
8	**4**	2.7	2	160	40	12
16	8	5.4	**4**	160	20	18

2.1 Switched-Resistor Preamplifier

The switched-resistor solution is illustrated in Figure 2.2, where R_{SW} is connected in feedback during phase ϕ_1 and to a reference voltage V_b in phase ϕ_2. This type of operation results in an equivalent resistance value of:

$$R_{F_{eq}} = R_{SW} \cdot \frac{S_L + S_R}{S_R} \tag{2.1}$$

that with $R_{SW} = 5$ MΩ and nominal values for S_L and S_R of 9950 ns and 50 ns, respectively, provides an equivalent resistance value of 1 GΩ. The multiplication factor $L = (S_L + S_R)/S_R$ can be programmed from 100 to 400 with steps of 100, so that $R_{F_{eq}}$ can vary from 0.5 GΩ to 2 GΩ with steps of 0.5 GΩ. Table 2.1 reports the values of the digital programmable capacitances to set the gain G from −6 dB to +18 dB with steps of 6 dB and the high-pass pole frequency f_{HP} for each value of L in every gain configuration.

The design of the switches is not critical for this preamplifier, since the switch in series to R_{SW} has a negligible on-resistance compared to R_{SW} itself, while the switches used for programming the variable capacitances are all connected to the preamplifier virtual ground, which exhibits negligible voltage swing. Since switches and capacitors are not an issue, the linearity of this preamplifier is limited by the closed-loop performance. To reduce the distortion coming from variation of the operational amplifier (opamp) closed-loop gain, R_{SW} is connected to the reference voltage V_B during phase ϕ_1.

In terms of noise performance the preamplifier is dominated by the opamp noise amplified by the factor $1 + C_I/C_F$. The feedback resistance contribution is negligible, considering that its noise is never sampled on the feedback capacitance, since the time constant of the

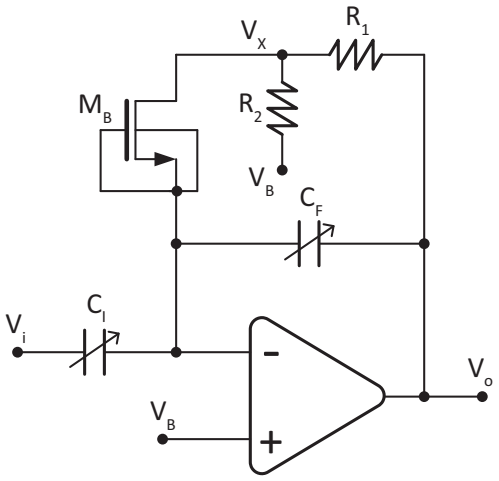


Figure 2.3: Off-transistor preamplifier

corresponding pole is much larger than S_R under any operating conditions.

2.2 Off-Transistor Preamplifier

Figure 2.3 shows the implementation of the preamplifier with an off-transistor M_B as feedback resistance. Transistor M_B is an NMOS device biased to guarantee $V_{GS} = 0$, designed with a channel width $W = 10$ μm and a channel length $L = 2$ μm. In this configuration, an extremely large impedance is achieved, providing a feedback loop always closed at *dc*. However, large negative signals at the preamplifier output could turn on M_B, reducing its impedance. In order to prevent this effect, a resistive divider composed by R_1 and R_2 has been introduced with an attenuation factor $R_2/(R_1 + R_2) = 1/1000$, thus guaranteeing that the voltage across M_B is always much smaller than its threshold voltage and, hence, M_B is actually off over the whole output signal swing.

From simulations, the equivalent feedback resistance turns out to be $R_F = 7$ GΩ, achieving the required low-frequency high-pass pole ($f_{HP} < 1$ Hz). The gain programmability is the same as in the switched-resistor implementation, with the same array of variable capacitances reported in Table 2.1.

Also for the off-transistor preamplifier the noise performance is dominated by the opamp noise amplified by the factor $1 + C_I/C_F$. The noise contributions coming from M_B, R_1, and R_2 are filtered out due to the extremely low value of the corresponding pole frequency f_{HP}. This value of f_{HP} leads also to high linearity, since the distortion contributions from

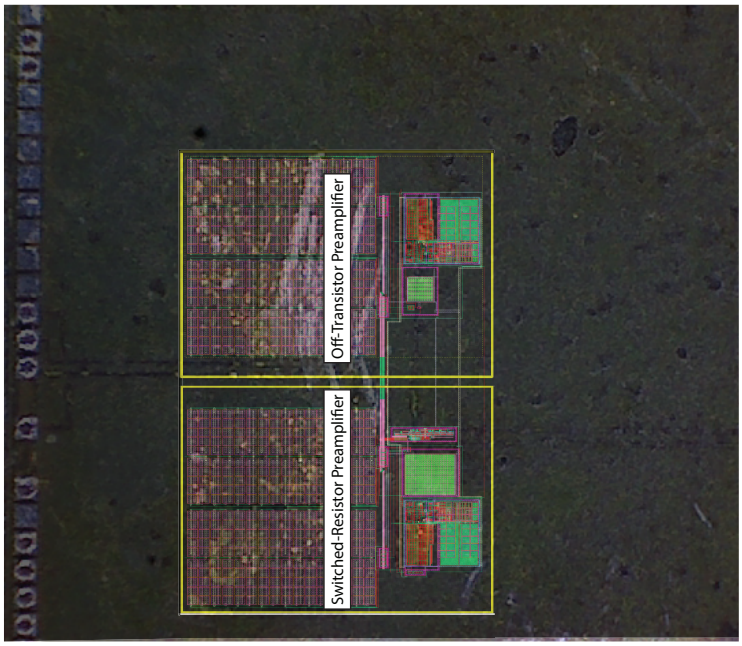


Figure 2.4: Microphotograph of the test chip containing both preamplifiers

M_B are also filtered.

2.3 Measurement Results

The two preamplifiers have been implemented in a standard 0.18-μm CMOS technology operating with a 1.8-V power supply voltage. The chip microphotograph is shown in Figure 2.4. There is no significant difference in area occupation (0.40 mm^2 each) between the switched-resistor and the off-transistor preamplifier, since the area is mainly dominated by the variable capacitances. The power consumption is 230 μW for both preamplifiers, since the opamp is the only active component.

Figure 2.5 shows the measured frequency response of the two preamplifiers in all of the gain configurations reported in Table 2.1. Both exhibit a cut-off frequency of the high-pass pole below 10 Hz. The larger feedback resistance implemented with the off-transistor results in a lower cut-off frequency than in the switched-resistor case ($L = 200$). A lower cut-off frequency ensures a larger safety margin for the high sensitivity to PVT variations of the off-transistor solution. The low-pass cut-off frequency is fixed by the opamp unity gain bandwidth and can be optimized based on the application requirements.

Linearity has been measured in terms of *THD* for an input tone at 5 kHz with -2 dB$_{\mathrm{FS}}$

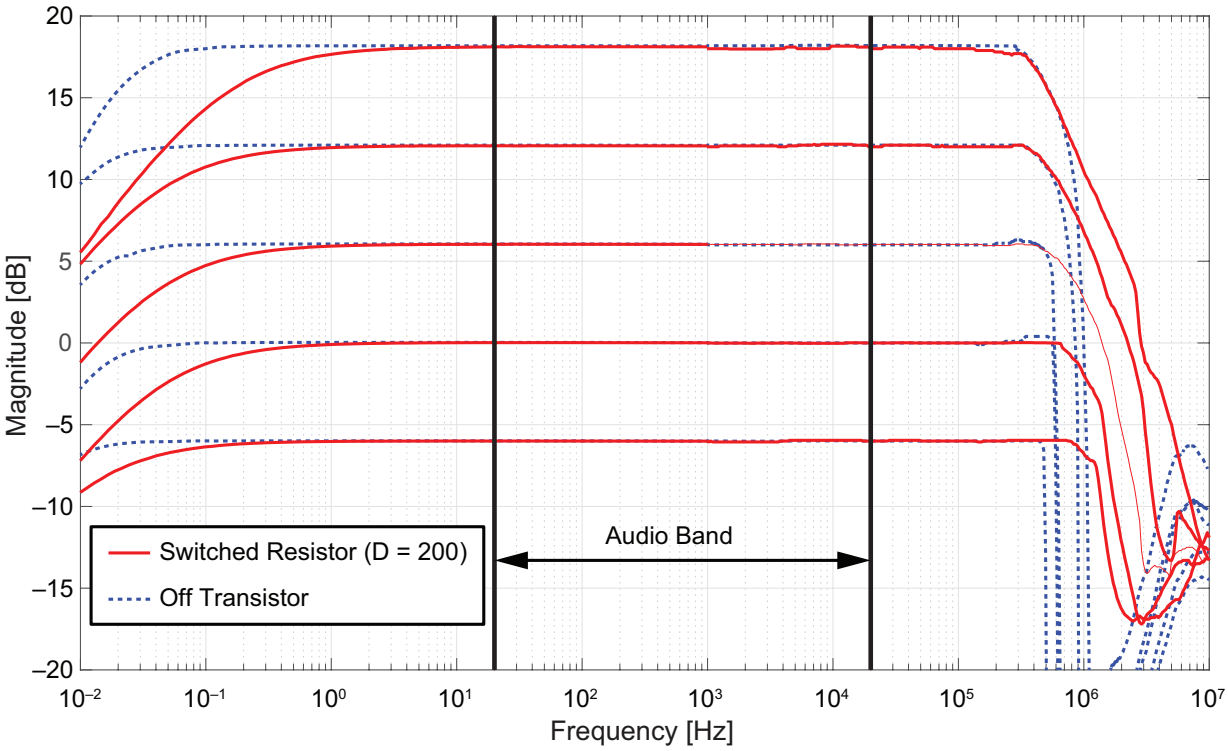


Figure 2.5: Frequency response of both preamplifiers in all of the gain configurations

amplitude. Figure 2.6 shows the achieved THD for both pre-amplifiers under these conditions as a function of the gain configuration. Both solutions achieve the target $THD < -100$ dB. The switched-resistor preamplifier achieves better linearity for all gain configurations. A more accurate measurement has been performed in both cases in a 0-dB gain configuration and for maximum capacitance values ($C_I = C_F = 160$ pF) with a -1 $\mathrm{dB_{FS}}$, 1-kHz input signal, achieving the results shown in Figure 2.7.

In terms of noise, the off-transistor preamplifier achieves slightly better performance than the switched-resistor preamplifier, as shown in Figure 2.8. However, with both solutions the input-referred noise ($IIRN$) remains under -100 dBV.

Finally, Figure 2.9 shows the simulations that have been carried out over temperature for the off-transistor in the range from $-20°$ C to $100°$ C to verify the behavior of the high-pass cut-off frequency. The pole frequency varies over three orders of magnitude, but it never exceeds 20 Hz.

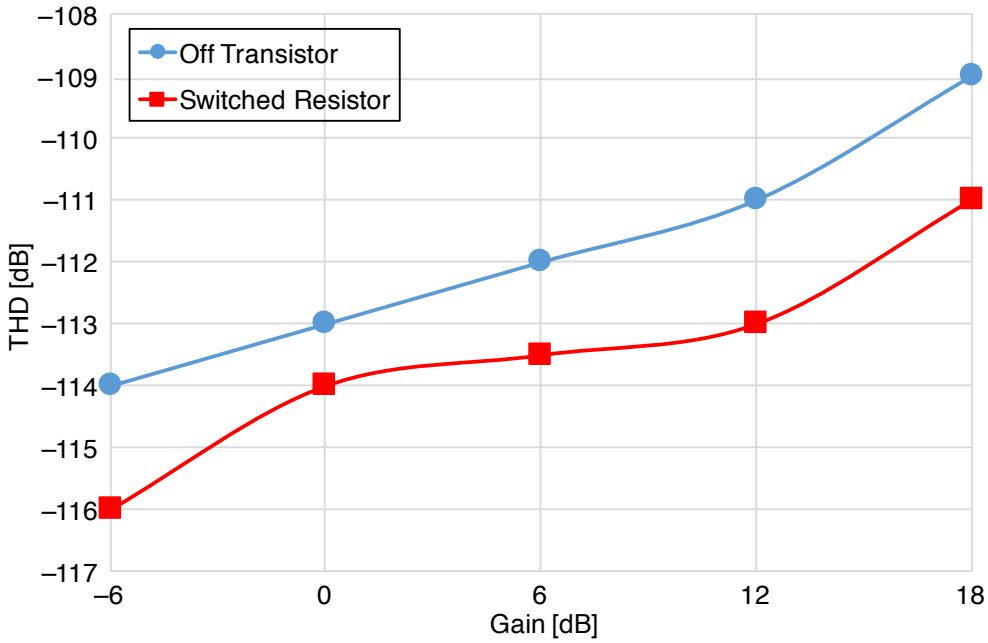


Figure 2.6: *THD* of both preamplifiers in all of the gain configurations

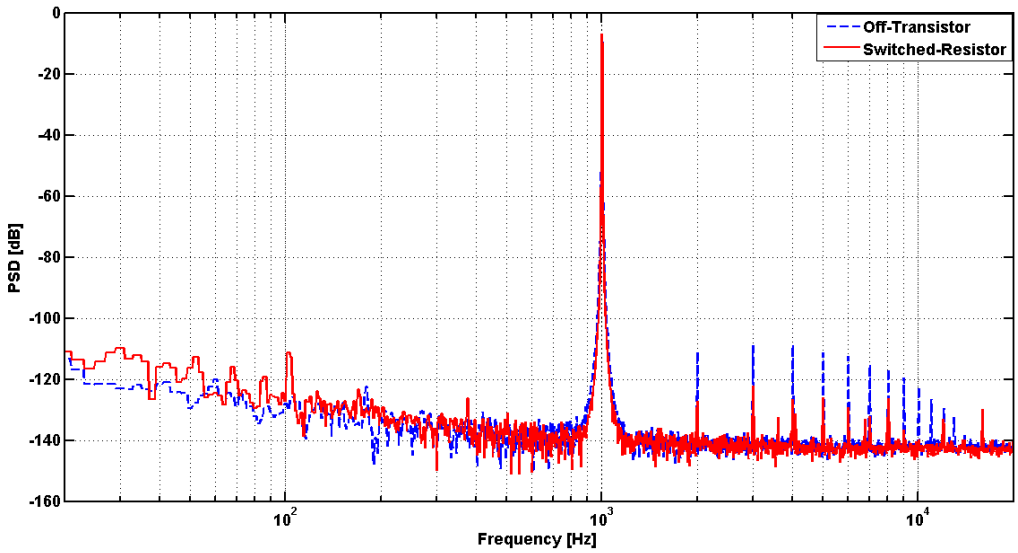


Figure 2.7: Spectrum of both preamplifiers for a 1-kHz,-1 $\mathrm{dB_{FS}}$ input signal in 0-dB gain configuration

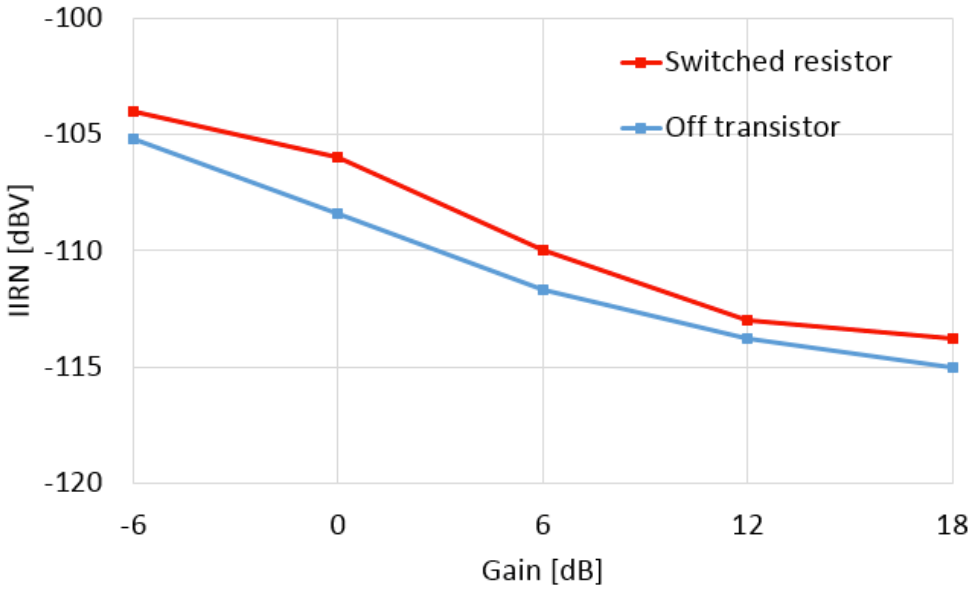


Figure 2.8: Input-referred noise of both preamplifiers in all of the gain configurations

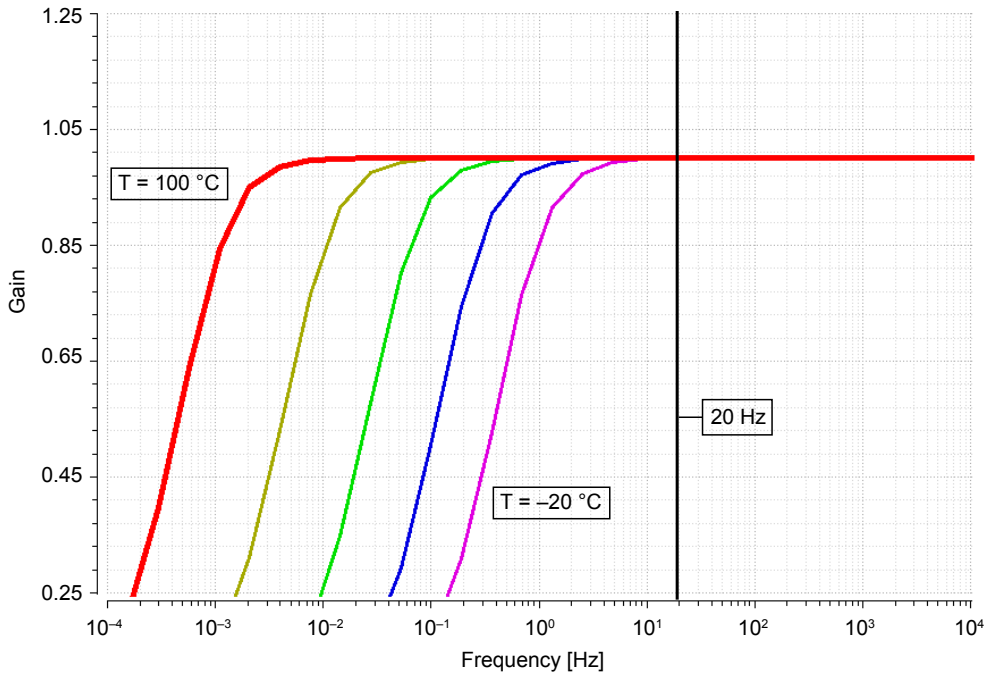


Figure 2.9: Simulated variations over PVT of the off-transistor preamplifier frequency response in 0-dB gain configuration

Chapter 3

Analog VAD Circuit

Most of the VAD techniques described in Chapter 1 are implemented in the digital domain, thus requiring the analog audio signal processing chain (SPC), including the A/D converter to be always operational. In order to reduce power, it would be much more efficient to implement the VAD circuit in the analog domain, thus allowing most of the blocks of the SPC to be maintained in power-down mode until the VAD circuit detects in the input signal coming from the microphone the presence of voice, as shown in Figure 3.1. Then, when voice is detected, the main part of the SPC is turned on, as shown in Figure 3.2.

Generally, the audio SPC is composed of a preamplifier stage with different gain configurations to provide the correct signal amplitude for the following A/D converter (ADC) and a digital signal processing unit, that performs more complex analysis on the incoming signal.

The idea behind this work is to implement a low-power solution that perform the signal

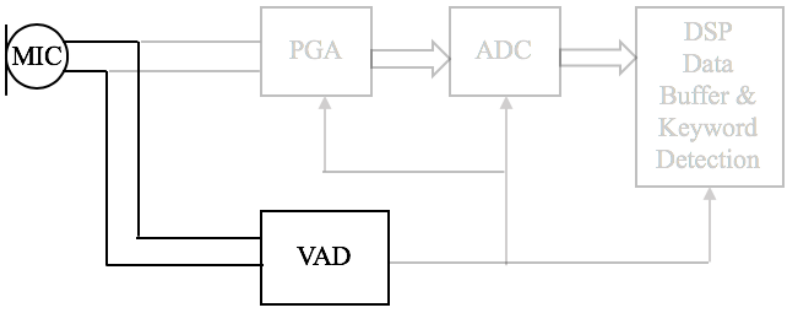


Figure 3.1: Signal processing chain in the absence of voice

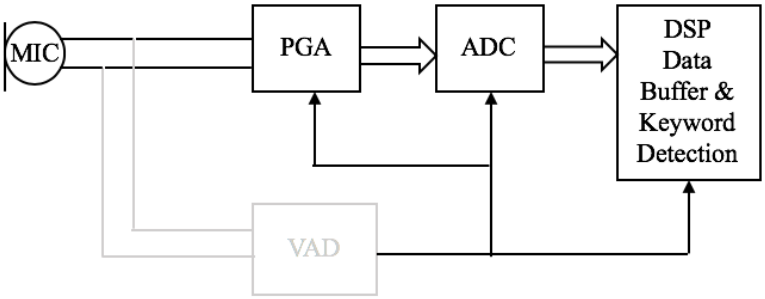
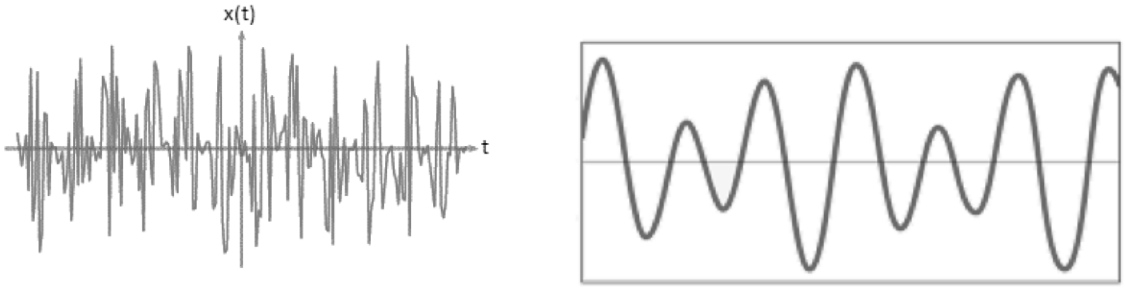


Figure 3.2: Signal processing chain in the presence of voice



(a) (b)

Figure 3.3: Typical audio input signal (a) and zoom over a frame of 16 ms (b)

processing algorithm for VAD with a completely analog circuit, where each processing stage extracts more complex information than the previous.

As discussed in Chapter 1, the most common VAD algorithms, typically implemented in the digital domain, are based on the evaluation of the energy $E(i)$ carried by the input signal $x(t)$ in a time frame of 16 ms (Figure 3.3)

$$E(i) = \int |x(t)|^2 dt \tag{3.1}$$

and on the assumption that, considering a long period of time, there are more unvoiced frames than voiced frames. This means that the energy evaluated for each frame can be considered as the environment background noise, which defines the noise level (NL) that will be used in the algorithm. The NL has to be updated at the end of every frame, making a comparison between the energy $E(i)$ of the current frame i and the noise level $NL(i-1)$

of the previous frame:

$$
\begin{aligned}
&NL(0) = \text{Initial Value} \\
&\textbf{\textit{if}}\ E(i) > NL(i-1) \\
&\quad NL(i) = \beta_1 \cdot NL(i-1) + (1-\beta_1) \cdot E(i) \qquad (3.2) \\
&\textbf{\textit{else}} \\
&\quad NL(i) = \beta_2 \cdot NL(i-1) + (1-\beta_2) \cdot E(i) \qquad (3.3)
\end{aligned}
$$

where β_1 and β_2 are coefficients varying from 0.95 to 0.995 with steps of 0.005. As can be observed in (3.2) and (3.3), the noise level tracks the energy with a faster rate for β_i values close to 0.95 and with a slower rate for β_i values close to 0.995, always keeping into account the noise reference value of the previous frame, to avoid sharp variations in the presence of sudden changes in the energy carried by the signal.

It is then possible to compare the signal-to-noise ratio (SNR), defined as

$$SNR(i) = \frac{E(i) - NL(i)}{NL(i)} \tag{3.4}$$

with a speech threshold (th_{SP}), to determine the presence or absence of speech in the audio signal ($SNR \geq th_{SP}$ in the presence of voice and $SNR < th_{SP}$ in the absence of voice), leading to:

$$\frac{E(i) - NL(i)}{NL(i)} \geq th_{SP} \tag{3.5}$$

The value of th_{SP} ranges from 0.1 to 5 with steps of 0.2. Since the division is not a straigth-forward operation to implement in the analog domain, (3.5) has been modified as follows:

$$
\begin{aligned}
E(i) &\geq NL(i) \cdot (1 + th_{SP}) \\
\frac{E(i)}{1 + th_{SP}} &\geq NL(i) \qquad (3.6)
\end{aligned}
$$

In this way the quantity $1/(1 + th_{SP})$ is always lower than one ([0.17:0.91]) and a simple energy averaging can be implemented.

Inside the analog VAD circuit of Figure 3.1, whose block diagram is shown in Figure 3.4, the audio signal coming from the microphone is processed by four stages. The first stage is a programmable-gain amplifier (PGA) that performs band-pass filtering and provides the correct amplitude and *dc* biasing for the second stage (see Chapter 2), where the square

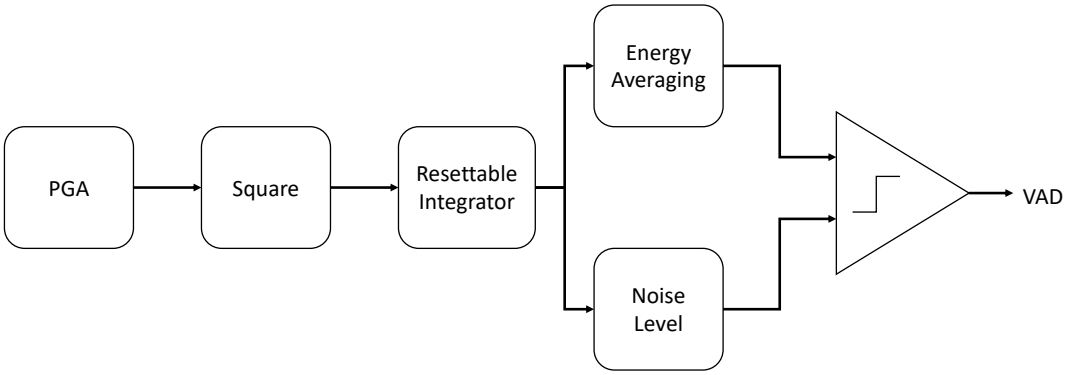


Figure 3.4: Block diagram of the analog VAD circuit

of the signal and a voltage to current conversion is performed. The output current of the square block is integrated through a resettable capacitance (current to voltage conversion) achieving the computation of the audio signal energy $E(i)$. At this point the processing chain is divided in two paths, one concerning the evaluation and update of NL and the other performing $E(i)$ averaging. In the last stage a dynamic latched comparator provides the VAD signal depending on the values of $E(i)$ and $NL(i)$.

3.1 Programmable-Gain Amplifier

The PGA is realized with a two stage fully-differential operational amplifier topology (Figure 3.5) with a common-mode feedback (CMF) circuit to set the output common-mode voltage (vcm) at the correct value for biasing the following stage. Two large 10-MΩ resistors, implemented with long-channel NMOS transistors, are used to determine the common-mode voltage vb_cm from the preamplifier output voltages von and vop, while transistor $M13$ closes the CMF loop generating the gate voltage for transistors $M4$ and $M5$. The total current consumption of this block is 300 nA, subdivided as follows:

- 50 nA in the first stage;
- 50 nA in the CMF circuit;
- 200 nA in the second stage.

With a supply voltage of 1.8 V, this leads to a power consumption of 540 nW.

The input signal of the PGA can be unbalanced differential (one PGA input is ac coupled to the signal source ground terminal) or balanced differential, in both cases the PGA is intended to be ac coupled to the microphone source with internal variable capacitors, as

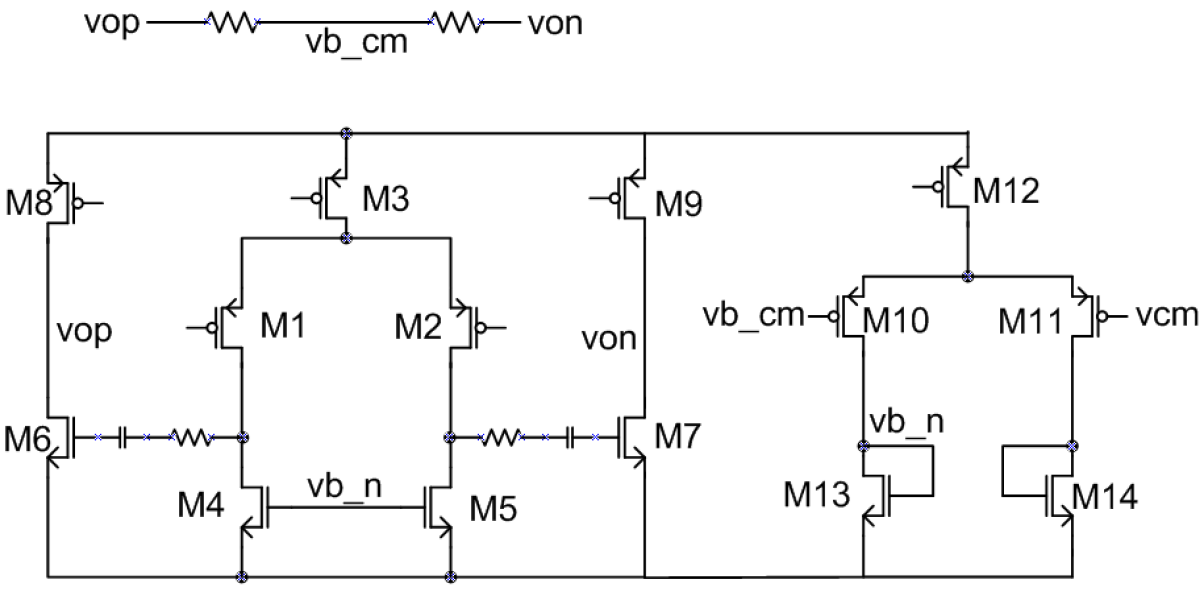


Figure 3.5: Schematic of the Miller opamp

shown in Figure 3.6 (see Chapter 2).

The gain of the PGA ranges from -12 dB to $+12$ dB with steps of 6 dB and it is set by the ratio of feedback resistance R_F over input resistance R_I. The choice of setting the transfer function gain with a resistance ratio and not with a capacitance ratio has been made to reduce the chip area, taking in account that MIM (metal-insulator-metal) capacitors will be used and in the layout they can be placed on top of resistors implemented with polysilicon. The PGA has a band-pass transfer function with a bandwidth of interest for VAD between 300 Hz and 6.8 kHz, to filter out unwanted high or low frequency components from the audio signal. The low frequency pole is achieved through the input RC network, while the high frequency pole is determined by the feedback RC network. The values of the components used are reported in Table 3.1. The feedback capacitance C_F in the higher gain configurations is constant. The reason for this decision comes from post-layout simulations, where picking was observed in the transfer function close to the high-frequency pole due to the parasitics capacitance of the resitors. The poly-resistor model, indeed, is shown in Figure 3.7, where C_P are the parasitic capacitances, R_C are the contact resistances, while the series of R_M is the designed resistance.

In spite of the use of MIM capacitors over poly-resistor, the PGA passive components occupy a large amount of area. To further reduce the area, the feedback resistance R_F

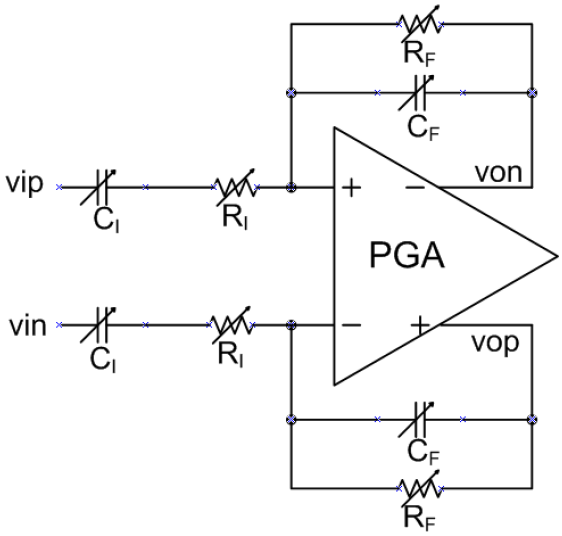


Figure 3.6: Feedback network around the PGA

Table 3.1: Gain programmability of the PGA (conventional resistor)

$\boldsymbol{Gain}$ [dB]	$\boldsymbol{C_I}$ [pF]	$\boldsymbol{C_F}$ [pF]	$\boldsymbol{R_I}$ [MΩ]	$\boldsymbol{R_F}$ [MΩ]
12	80	1.9	6.5	26
6	48	1.9	10.83	21.67
0	32	1.9	16.25	16.25
−6	24	2.72	21.67	10.83
−12	20	4.48	26	6.5

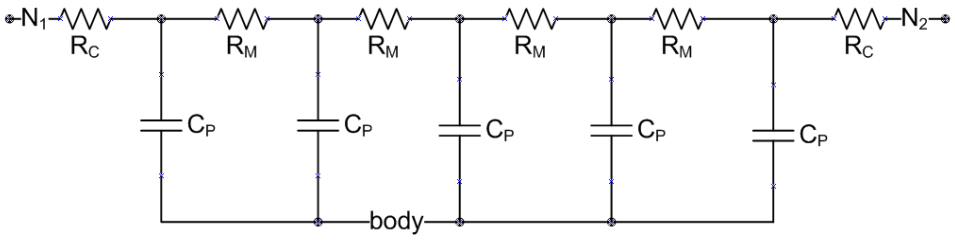
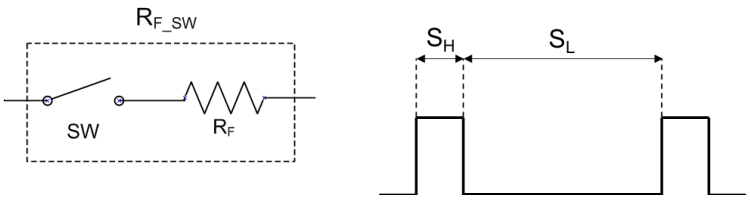


Figure 3.7: Poly-resistor model

Table 3.2: Gain programmability of the PGA (switched resistor)

$\boldsymbol{Gain}$ [dB]	$\boldsymbol{C_I}$ [pF]	$\boldsymbol{C_F}$ [pF]	$\boldsymbol{R_I}$ [MΩ]	$\boldsymbol{R_F}$ [MΩ]	$\boldsymbol{R_{F\_SW}}$ [MΩ]
12	16	4	1.46	2.18	133.8
6	16	8	1.46	1.09	66.9
0	8	8	2.93	1.09	66.9
−6	8	16	2.93	0.55	33.4
−12	4	16	5.85	0.55	33.4



(a) (b)

Figure 3.8: Switched-resistor implementation (a) and clock signals (b)

can be implemented with an off transistor or with a switched resistor, as discussed in Chapter 2. We decided to adopt the switched-resistor solution (Figure 3.8), due to the small bandwidth of interest and to the large variation over temperature of the off-transistor equivalent resistance, that would introduce a large variability of the pole frequency. The gain configurations for the switched-resistor solution are summarized in Table 3.2, where a substantial reduction of capacitance and resistance can be observed: from 84.48 pF to 32 pF capacitance per branch and from 32.5 MΩ to 6.4 MΩ resistance per branch. The equivalent value of the switched resistance R_{F_SW} depends on the clock period and the duty cycle:

$$R_{F\_SW} = R_F \cdot \frac{S_H + S_L}{S_H}; \tag{3.7}$$

where S_H is the phase where switch SW is closed and R_F is connected to the PGA output, while during phase S_L SW is open and R_F disconnected from the PGA output. Phase S_H has been implemented with half period of the input clock at 3.068 MHz and the switching period is 10 μs, leading to a multiplication factor $(S_H + S_L)/S_H$ approximately equal to 61.

The open-loop gain and phase without and with load connected to the PGA outputs are reported in Figure 3.9 and Figure 3.10, respectively. The load consists of the feedback

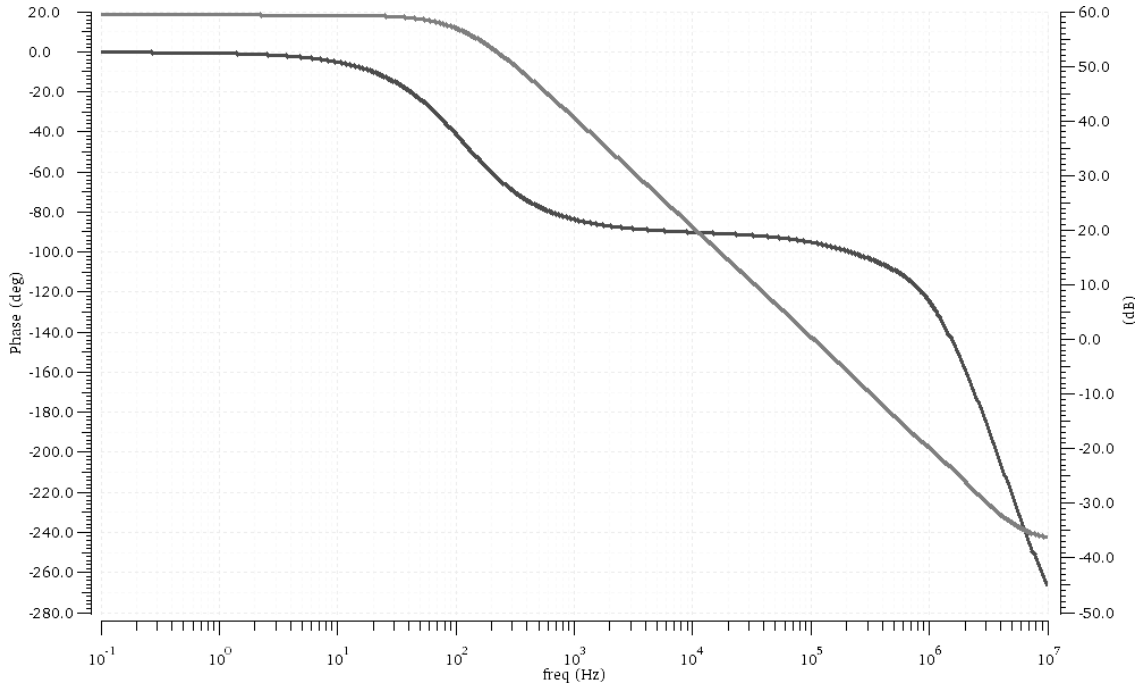


Figure 3.9: Simulated pen-loop gain and phase of the PGA with no load

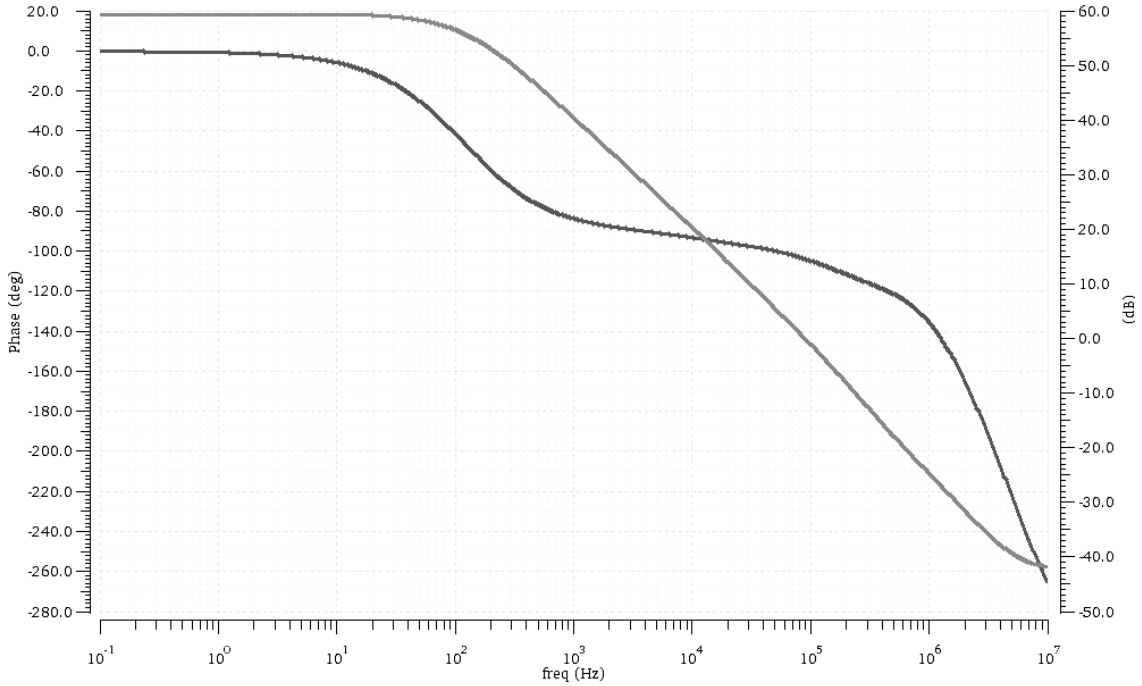


Figure 3.10: Simulated open-loop gain and phase of the PGA with load

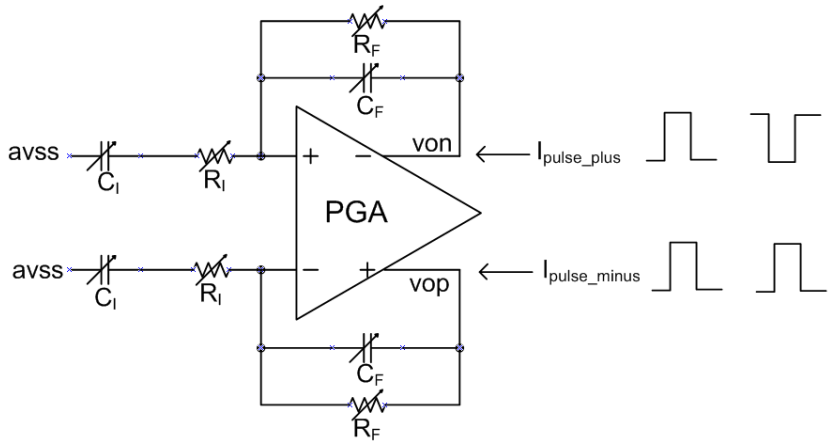


Figure 3.11: Circuit used to verify the PGA stability in transient conditions

and input *RC* networks connected between PGA outputs and ground in the highest gain configuration. The PGA has a *dc* open-loop gain of 58 dB with 100 kHz unity-gain bandwidth, a phase margin of 89° and gain margin of 30 dB in the configuration with no load and a phase margin of 104° and gain margin of 34 dB in the configuration with load applied.

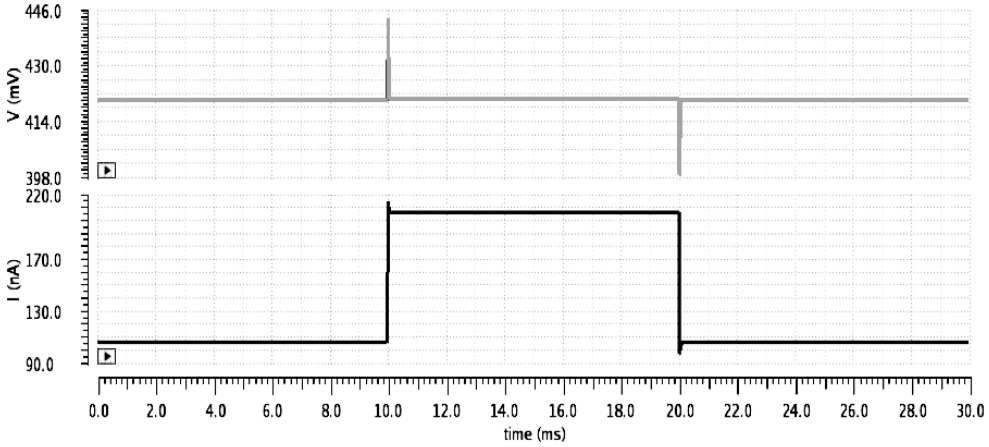
To further check the stability of the PGA, 100-nA common-mode and differential current pulses whit pulse width of 10 ms have been injected at the output nodes, as shown in Figure 3.11, monitoring *von* and *vop* for oscillations. The results, shown in Figure 3.12 and Figure 3.13, confirm that no oscillations occur.

3.2 Signal Energy Computation

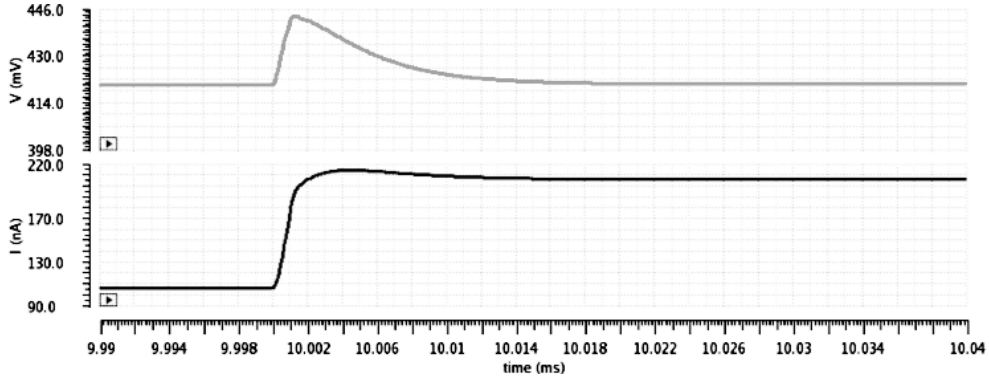
In order to implement the VAD algorithm it is necessary to compute the energy of the signal. In this section will be explained how the energy carried by the signal has been extrapolated. According to (3.1) and Figure 3.4, the energy computation is the sequence of two operations:

- Calculation of the signal square;
- Integration of the result of the audio frame (16 ms).

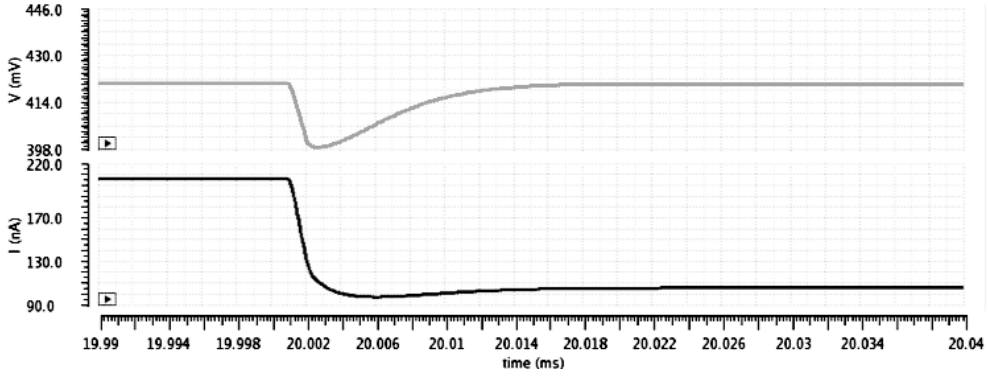
Implementing the square operation in the analog domain is not a straight-forward task. A possible solution is to exploit the quadratic relation between current and voltage in a MOS



(a)

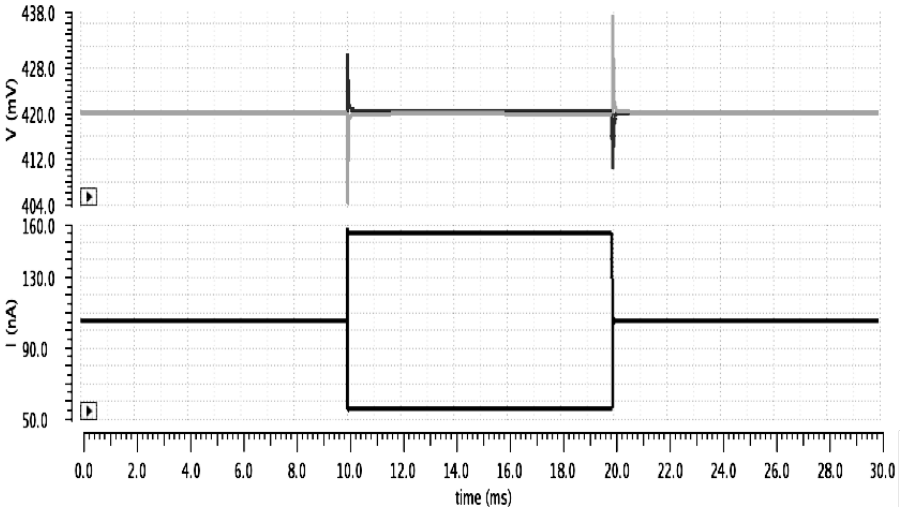


(b)

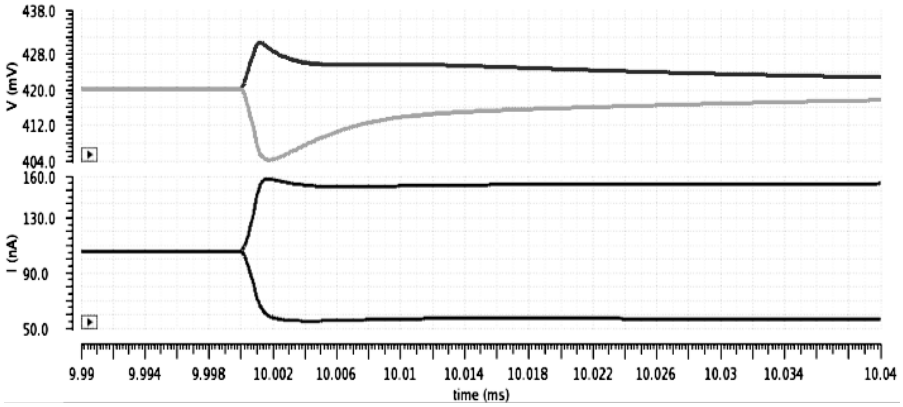


(c)

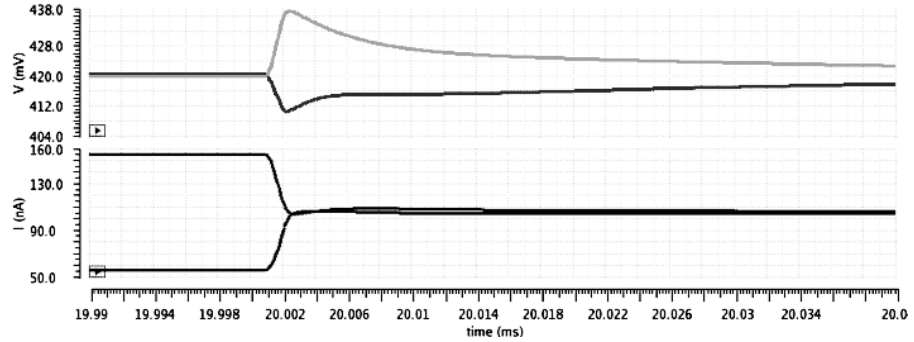
Figure 3.12: Transient simulation to verify stability with common-mode current pulses applied at the output of the PGA



(a)



(b)



(c)

Figure 3.13: Transient simulation to verify stability with differential current pulses applied at the output of the PGA

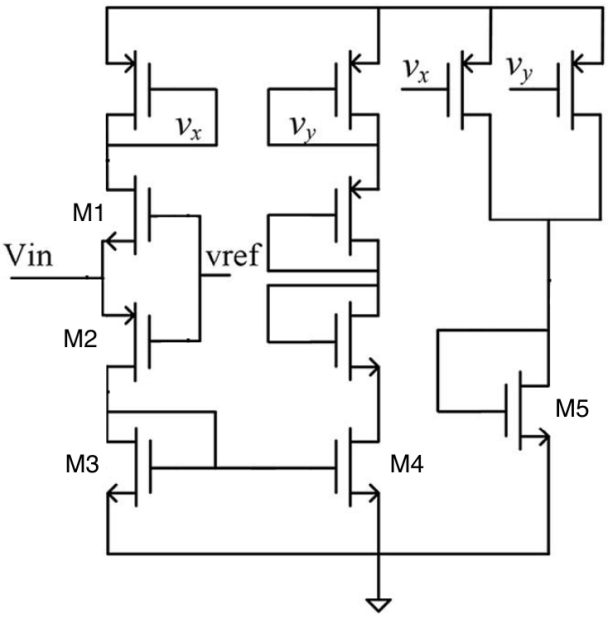


Figure 3.14: Basic idea for implementing the square operation [5]

transistor:

$$I_D = k \cdot \left(|V_G| - |V_S| - |V_{th}|\right)^2 \tag{3.8}$$

where I_D is the drain current current, k is the constant related to mobility, oxide capacitance, and transistor dimensions, V_G, V_S, and V_{th} are the gate, source, and threshold voltages, respectively.

The signal can be applied at the source or at the gate of the transistor that has to perform the square, keeping the other voltages constant. An implementation of this idea, described in [5], is shown in Figure 3.14, where the input signal Vin is applied to the source of either an NMOS and a PMOS transistor ($M1$ and $M2$), while their gates are connected to a fixed reference voltage $vref$. When the input signal is larger than the reference ($Vin > vref$), $M2$ performs the square and the resulting current is collected by the output transistor $M5$ through the current mirrors. On the other hand, if the input signal is lower than the reference ($Vin < vref$), $M1$ performs the square and the resulting current is again delivered to $M5$.

The solution adopted in this work, whose schematic is shown in Figure 3.15, is based on the same principle. In this case, the square is implemented through the a couple of NMOS transistors, whose sources are connected to ground, while their gates are connected to PGA differential output. The *dc* common-mode output voltage of the PGA has been designed

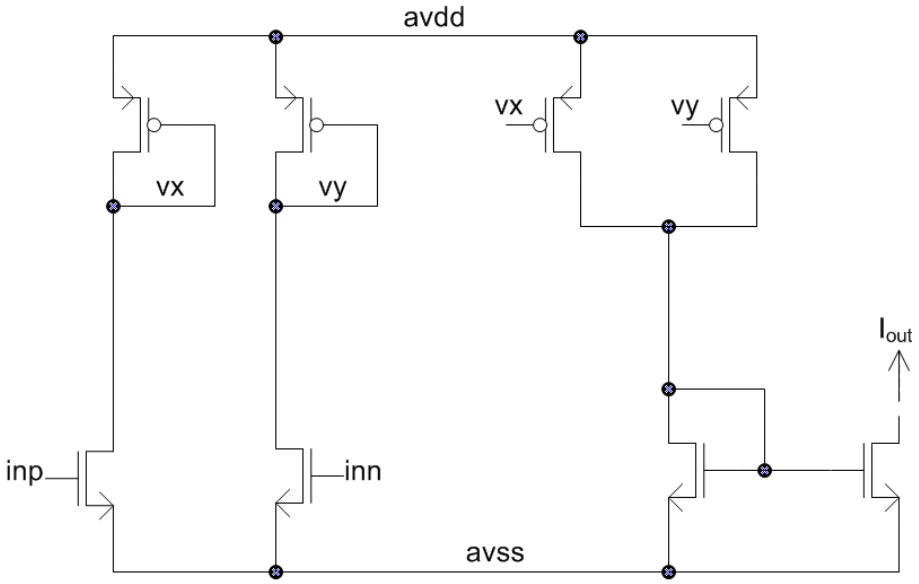


Figure 3.15: Adopted circuit for implementing the square operation

to track the NMOS transistor threshold voltage variations, thus avoiding large variations of the transistor current. Indeed, in order to achieve low power consumption, which is a fundamental requirement for this circuit, the static current consumption has to be as low as possible. Therefore, the input transistors have been designed with large L and small W leading to a total current consumption of 18 nA. The voltage of the common-mode comes from a transistor diode connected, in this way the pseudo differential pair behaves as a current mirror which current variations are mainly determined by the signal coming from the preamplifier.

The operation of the signal squarer is illustrated in Figure 3.16, where the square of a differential input ramp signal centered around the PGA output common-mode voltage is computed and converted from differential to signle-ended. Figure 3.16 shows both the ideal squared waveform and the waveform obtained with the implemented circuit.

The next step to obtain the audio signal energy is the integration of the output current of the squarer circuit, achieved through the resettable integrator shown in Figure 3.17, which implements the function:

$$V_{out\_int} = \int_{t_i}^{t_f} \frac{I_{out\_squarer}}{C_{INT}} dt; \tag{3.9}$$

where V_{out_int} is the integrator output voltage that represents the signal energy. The

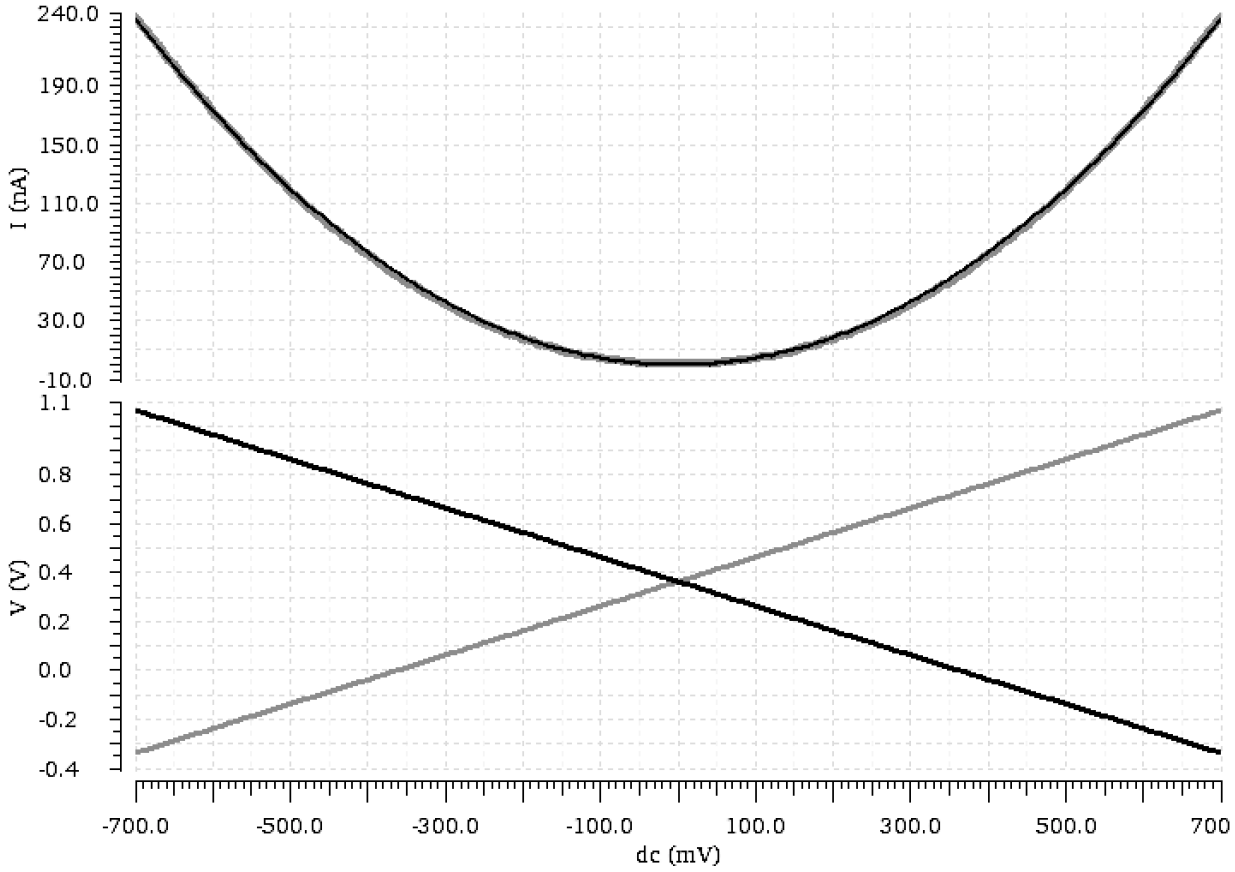


Figure 3.16: Squarer output current and ideal square waveform overlapped (top) obtained with a ramp input signal (bottom) implemented with a DC sweep.

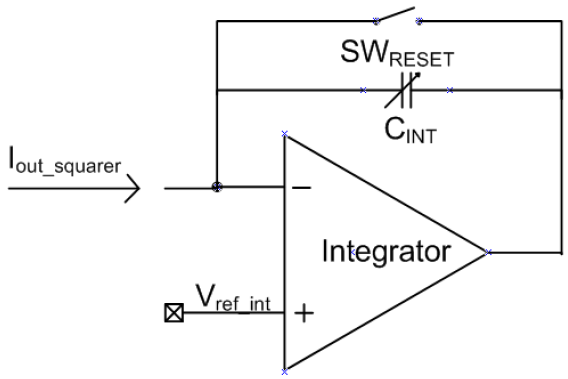


Figure 3.17: Resettable integrator

integration capacitance (C_{INT}) and period are programmable. The possible values of C_{INT} are 10 pF, 20 pF, or 40 pF, while the integration period can be of 8 ms, 16 ms, or 32 ms. Using (3.9) it is possible to estimate the voltage swing required for V_{out_int}. For example, a sinusoidal current $I_{out\_squarer}$ with 1-nA amplitude at 1 kHz with an integration period of 16 ms and an integrating capacitance of 20 pF leads to an integrator output variation of 400 mV:

$$V_{out\_int} = \frac{1 \text{ nA}}{20 \text{ pF}} \int_0^{0.016} \sin^2(2\pi t \cdot f_{1 \text{ kHz}}) dt = 0.4 \text{ V} \tag{3.10}$$

Figure 3.18 represents the integrator output when the squarer inputs are biased at the preamplifier common-mode level and no other signal is applied. The final integration value is about 60 mV, which corresponds to an input current of 2.4 pA (integrator period of 16 ms and $C_{INT} = 40$ pF).

The integrator amplifier, with a load of 40 pF, features a *dc* open-loop gain of 60 dB with 2.2-kHz unity gain bandwidth, as shown in Figure 3.19.

Figure 3.20 shows the biasing circuit for the PGA, the squarer, and the integrator. This circuit generates the biasing current for the PGA (*ibias_preamp*), the PGA common-mode voltage reference (*vbias_cm_preamp*), that tracks the variations of the NMOS transistor threshold voltage, and the biasing current for the integrator (*ibias_int*).

The overall schematic of PGA, squarer, and integrator is summarized in Figure 3.21. The output of this circuit represents the energy of the audio signal.

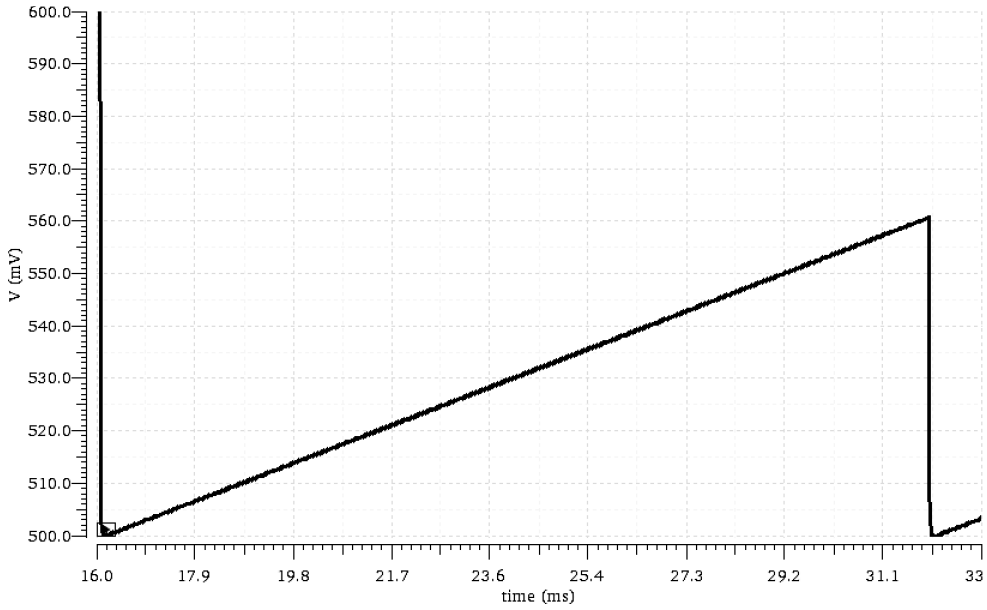


Figure 3.18: Integrator output offset

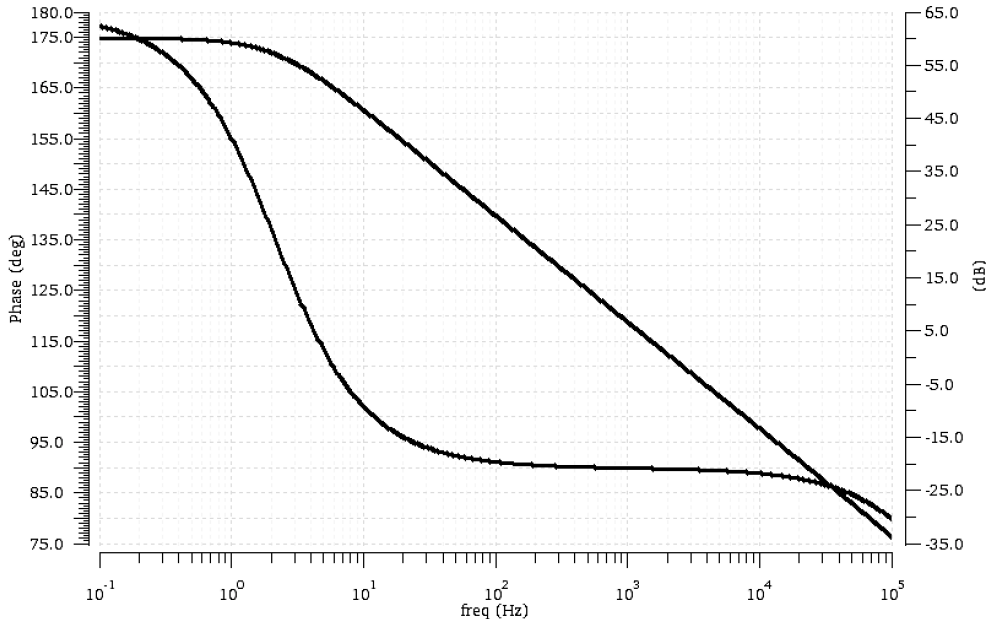


Figure 3.19: Integrator open-loop gain and phase with 40-pF load

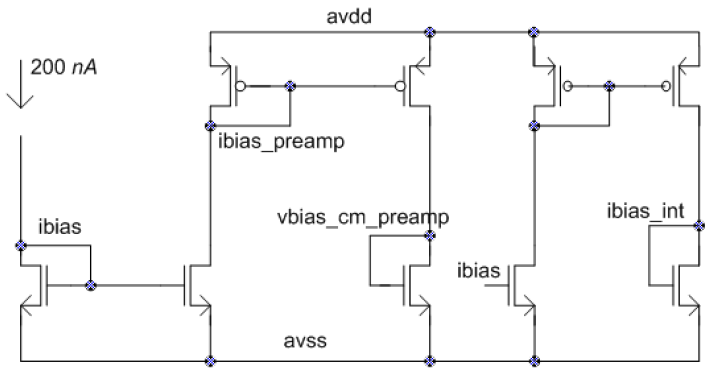


Figure 3.20: Biasing circuit for the PGA, the squarer, and the integrator

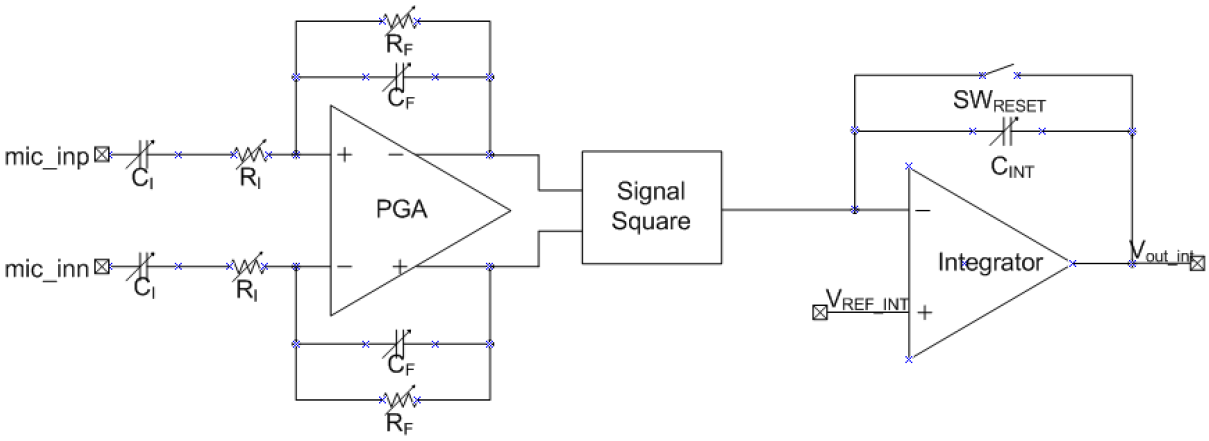


Figure 3.21: Overall schematic of PGA, squarer, and integrator

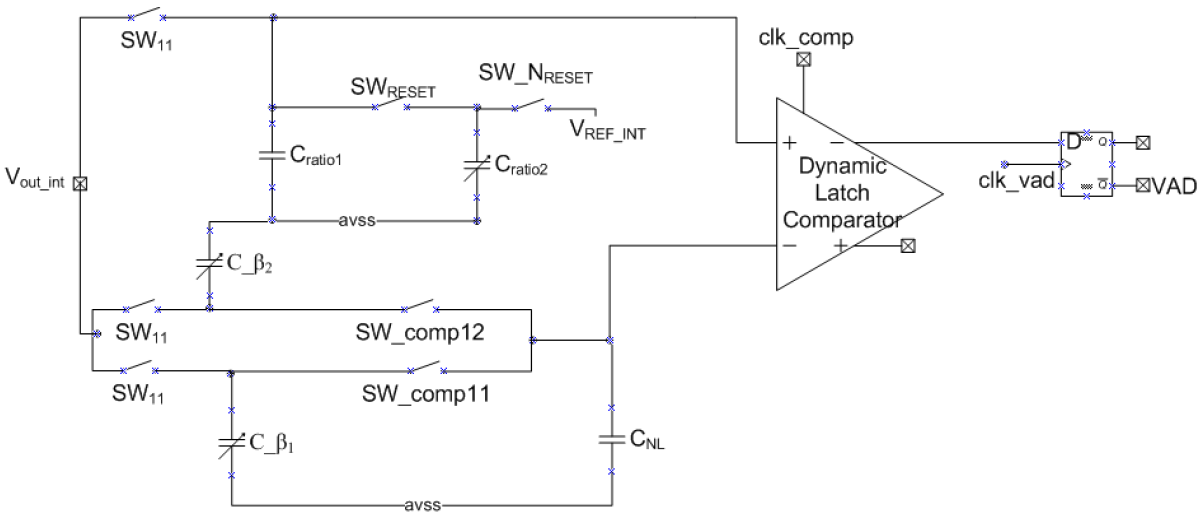


Figure 3.22: Schematic of the circuit used for VAD generation

3.3 Energy Averaging and Noise Level Computation

The energy of the audio signal obtained at the output of the circuit shown in Figure 3.21 has now to be processed to obtain the final VAD output, according to (3.2), (3.3), and (3.6). The schematic of the circuit used for realizing these functions is shown in Figure 3.22.

The integrator output V_{out_int} enters into two parallel paths, one devoted to the averaging of the energy and the other to the evaluation of the noise level NL, which represent the inputs of the comparator used to produce the final VAD signal. Before going through the functional description of this circuit, referring to Figure 3.22, some quantities have to be defined:

- $E(i)$ is the energy of the input signal evaluated over integration period i and given by V_{out_int};
- $NL(i)$ is the noise level in integration period i stored on the fixed capacitance C_{NL};
- C_β_1 and C_β_2 are the variable capacitances used to update the noise level;
- SW_comp11 and SW_comp12 are the switches that select with which capacitance (C_β_1 or C_β_2) the noise level has to be updated;
- C_{ratio1} is the fixed capacitance where the energy $E(i)$ is stored;
- C_{ratio2} is the variable capacitance used to average the energy $E(i)$;
- SW_{RESET} is the switch used to reset the integrating capacitance and average the energy;

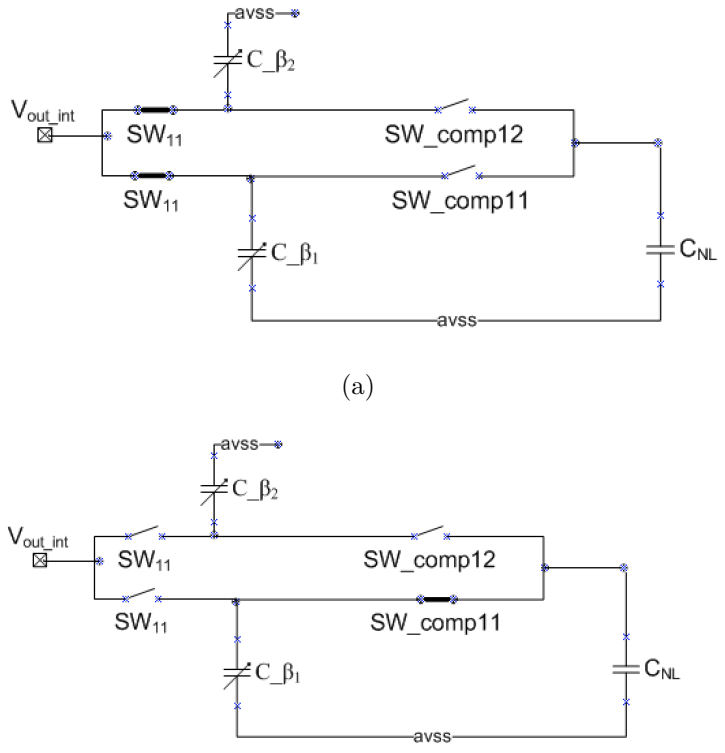


Figure 3.23: NL update circuit during the integration period (a) and during the update period with C_β_1 (b)

- SW_N_{RESET} is the switch used to reset C_{ratio2};
- SW_{11} is the switch that sets the integration period.

As stated previously, the noise level is a measurement of the background noise and it has to be updated every integration period with a fraction of the energy $E(i)$. The algorithm used to update NL is described by (3.2) and (3.3), which are implemented by the bottom part of the circuit shown in Figure 3.22. During the integration time both C_β_1 and C_β_2 are connected to the integrator output, while on C_{NL} is held the noise level value from the previous integration period (Figure 3.23a), leading to the following charge distribution:

$$Q_{NL}(0) = E(i) \cdot (C\_\beta_1 + C\_\beta_2) + NL(i-1) \cdot C_{NL} \tag{3.11}$$

At the end of the integration time switches SW_{11} are opened and, depending on the comparison between $E(i)$ and $NL(i-1)$, one between SW_comp11 and SW_comp12 is closed to obtain the updated value of $NL(i)$. In particular, switch SW_comp11 is closed

Table 3.3: Possible values of C_β_1 and C_β_2

$\boldsymbol{\beta}$	0.95	0.955	0.96	0.965	0.97	0.975	0.98	0.985	0.99	0.995
$\boldsymbol{C\_\beta_i}$ **[pF]**	0.5	0.45	0.4	0.35	0.3	0.25	0.2	0.15	0.1	0.05

when the new noise level has to be increased (Figure 3.23b), while SW_comp12 is closed when the new noise level has to be decreased:

$$
\begin{aligned}
&E(i) > NL(i-1)\\
&\quad Q_{NL}(1) = NL(i)\cdot(C_{NL}+C\_\beta_1) + E(i)\cdot C\_\beta_2\\
&\quad NL(i)\cdot(C_{NL}+C\_\beta_1) = E(i)\cdot C\_\beta_1 + NL(i-1)\cdot C_{NL}\\
&\quad NL(i) = E(i)\cdot\frac{C\_\beta_1}{C\_\beta_1+C_{NL}} + NL(i-1)\cdot\frac{C_{NL}}{C_{NL}+C\_\beta_1}
\end{aligned}
\tag{3.12}
$$

$$
\begin{aligned}
&E(i) \le NL(i-1)\\
&\quad Q_{NL}(1) = NL(i)\cdot(C_{NL}+C\_\beta_2) + E(i)\cdot C\_\beta_1\\
&\quad NL(i)\cdot(C_{NL}+C\_\beta_2) = E(i)\cdot C\_\beta_1 + NL(i-1)\cdot C_{NL}\\
&\quad NL(i) = E(i)\cdot\frac{C\_\beta_2}{C\_\beta_2+C_{NL}} + NL(i-1)\cdot\frac{C_{NL}}{C_{NL}+C\_\beta_2}
\end{aligned}
\tag{3.13}
$$

From (3.12) and (3.13) it is easy to extrapolated that:

$$
\beta_1 = \frac{C_{NL}}{C_{NL}+C\_\beta_1} \qquad \beta_2 = \frac{C_{NL}}{C_{NL}+C\_\beta_2} \tag{3.14}
$$

Knowing that the minimum capacitance that can be implemented in the adopted technology is 50 fF and that the range of values for β_1 and β_2 is [0.95:0.005:0.995], C_{NL} has been implemented with a value of 10 pF, while C_β_1 and C_β_2 are two identical arrays of ten capacitors with unit value of 50 fF, as summarized in Table 3.3.

Energy averaging is performed by the upper part of the circuit shown in Figure 3.22, based on the following equation:

$$
E_{avg}(i) = E(i)\cdot\frac{1}{1+th_{SP}} \tag{3.15}
$$

where $E(i)$ is the energy stored on C_{ratio1} during the integration period and th_{SP} is a coefficient ranging from 0.1 to 5 with steps of 0.2. In the first phase SW_{11} and SW_N_{RESET}

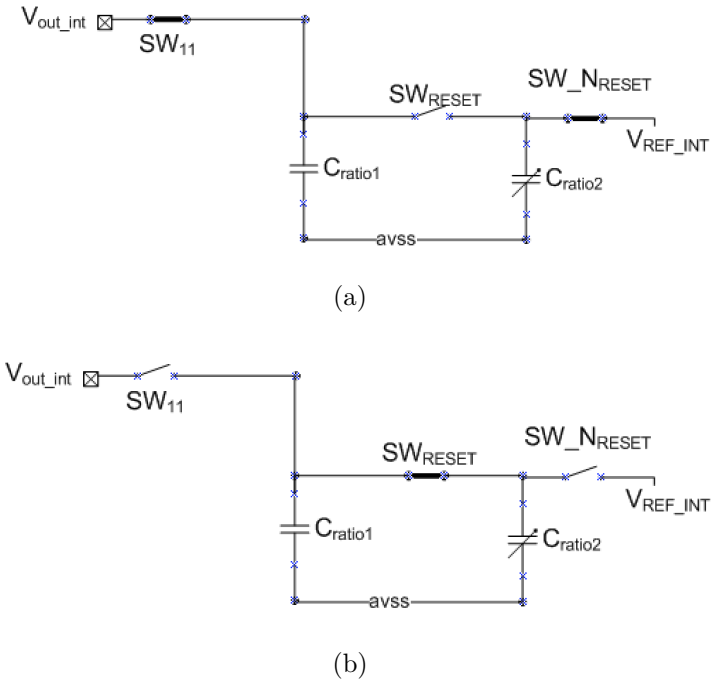


Figure 3.24: Energy averaging circuit during the integration period (a) and during VAD decision (b)

are closed, as shown in Figure 3.24a, and charge distribution occurs according to:

$$Q_{avg}(0) = V_{out\_int} \cdot C_{ratio1} + V_{REF\_INT} \cdot C_{ratio2} \tag{3.16}$$

After the comparator decision to select the C_β_i capacitance to use, SW_N_{RESET} is opened and SW_{RESET} is closed to perform the averaging with the variable capacitance C_{ratio2}, according to:

$$Q_{avg}(1) = V_F \cdot (C_{ratio1} + C_{ratio2}) \tag{3.17}$$

From (3.16) and (3.17) it is possible to evaluate the output voltage of the averaging circuit:

$$V_F = V_{out\_int} \cdot \frac{C_{ratio1}}{C_{ratio1} + C_{ratio2}} + V_{REF\_INT} \cdot \frac{C_{ratio2}}{C_{ratio1} + C_{ratio2}} \tag{3.18}$$

In (3.18) it seems that also V_{REF_INT}, which is the staring point of the integration, is averaged, thus introducing an offset that varies depending on the selected value of C_{ratio2}. However, actually, V_{out_int} can be split in two components:

$$V_{out\_int} = V_{int} + V_{REF\_INT} \tag{3.19}$$

where V_{REF_INT} is constant, while V_{int} is the effective energy of the input signal. Therefore, (3.18) can be re-written as:

$$V_F = V_{int} \cdot \frac{C_{ratio1}}{C_{ratio1} + C_{ratio2}} + V_{REF\_INT} \tag{3.20}$$

Comparing (3.15) and (3.20), we obtain:

$$\frac{C_{ratio1}}{C_{ratio1} + C_{ratio2}} = \frac{1}{1 + th_{SP}} \tag{3.21}$$

Table 3.4 summarizes the values available for the variable capacitance C_{ratio2} for $C_{ratio1} =$

Table 3.4: Possible values of C_{ratio2}

$\frac{1}{1+th_{SP}}$	0.91	0.88	0.85	0.82	0.79	0.76	0.72	0.69	
$\boldsymbol{C_{ratio2}}$ **[pF]**	0.5	0.69	0.89	0.1.12	1.36	1.62	1.9	2.2	
$\frac{1}{1+th_{SP}}$	0.66	0.63	0.6	0.57	0.54	0.51	0.48	0.45	
$\boldsymbol{C_{ratio2}}$ **[pF]**	2.54	2.91	3.31	3.76	4.26	4.82	5.45	6.17	
$\frac{1}{1+th_{SP}}$	0.42	0.39	0.35	0.32	0.29	0.26	0.23	0.2	0.17
$\boldsymbol{C_{ratio2}}$ **[pF]**	7	7.96	9.08	10.42	12.05	14.05	16.58	19.9	24.41

5 pF. The programmability step is not fixed as for C_β_1 and C_β_2, but changes to obtain a constant step in the ratio given by (3.21).

Figure 3.25 summarizes the circuit clock phases at the end of the integration period. Switches SW_{11} and SW_N_{11} ($SW\_N_{11} = \overline{SW_{11}}$) determine the integration period. The circuit configuration during the integration period is shown in Figure 3.26a, where C_{ratio1}, C_β_1, and C_β_2 are charged with the integrator output voltage, C_{ratio2} is reset to V_{REF_INT}, and C_{NL} holds the noise level from the previous period $NL(i-1)$. At the end of the integration period switches SW_{11} are opened Figure 3.26 and the comparator takes the first decision (*clk_comp*), comparing the energy obtained $E(i)$ with $NL(i-1)$. The comparator is then reset, capacitance C_{ratio2} is connected to C_{ratio1}, and one between SW_comp11 and SW_comp12 is closed, according to:

$$E(i) > NL(i-1) \quad \Longrightarrow \quad SW\_comp11$$
$$E(i) \leq NL(i-1) \quad \Longrightarrow \quad SW\_comp12$$

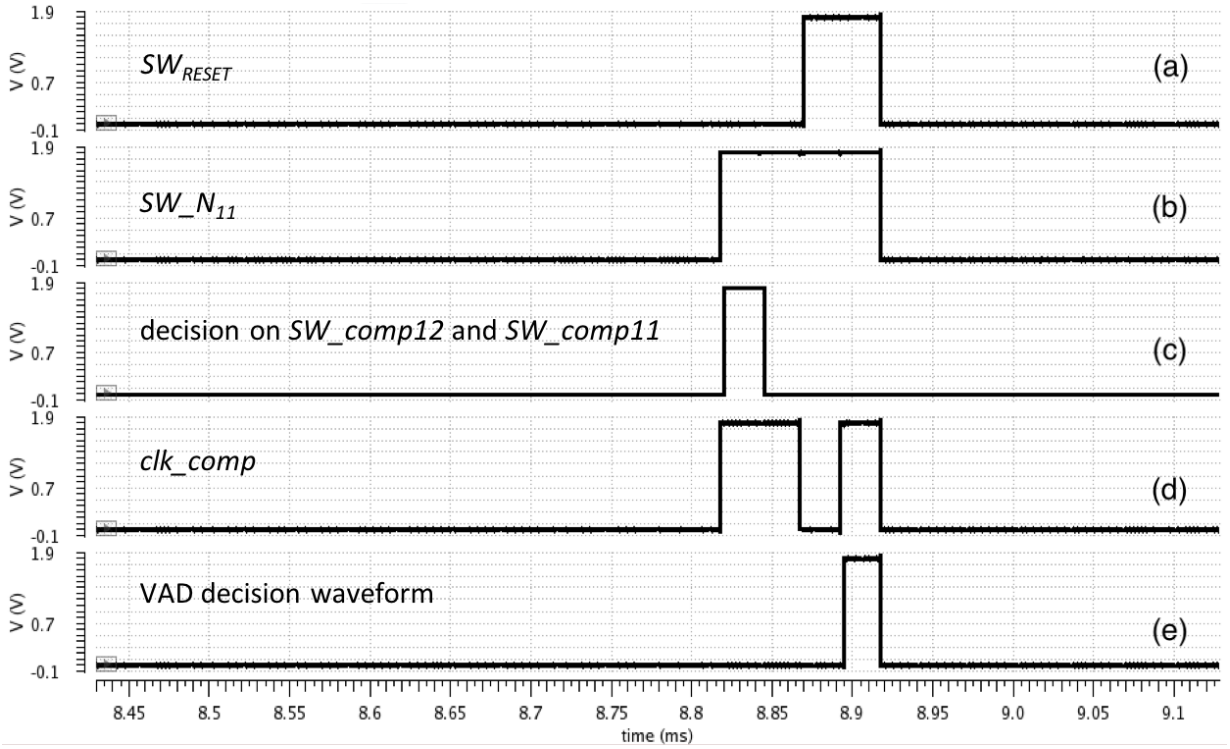
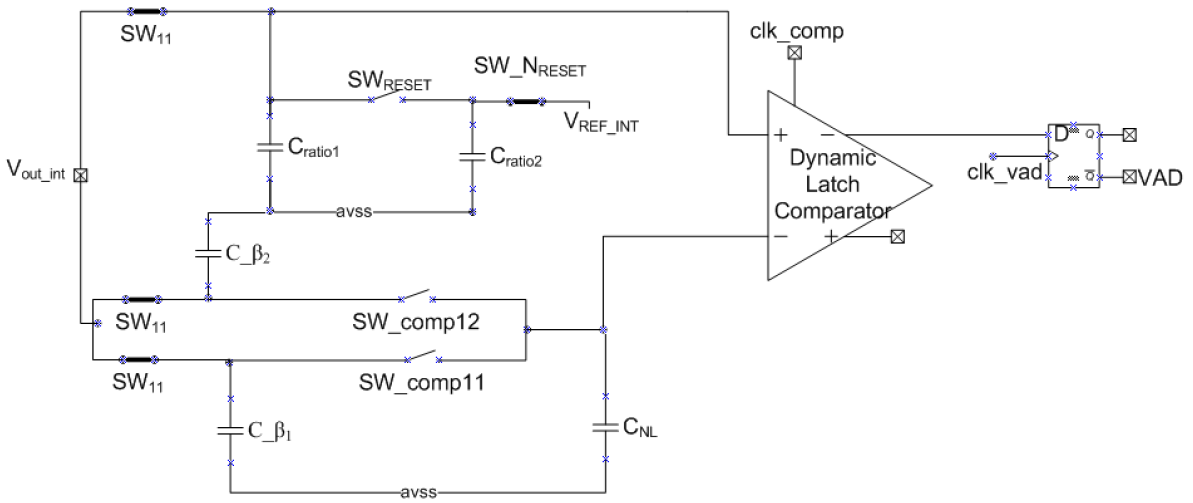
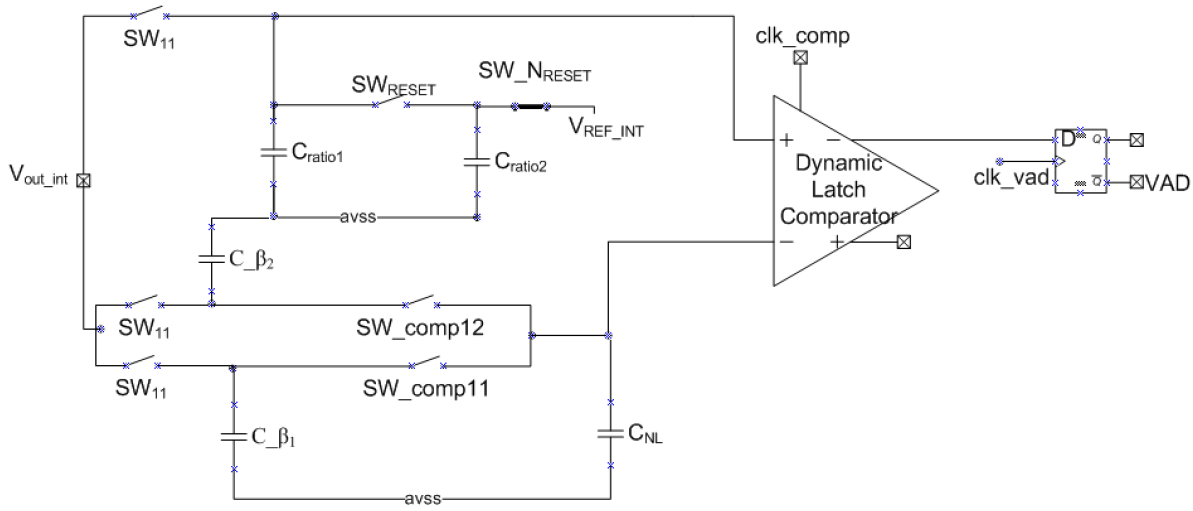


Figure 3.25: Clock phases: integrator capacitance reset (a), evaluation phase (b), selection between SW_comp11 and SW_comp12 (c), comparator clock (d), VAD decision (e)



(a)



(b)

Figure 3.26: Circuit configuration during the integration period (a) and during the first comparison (b)

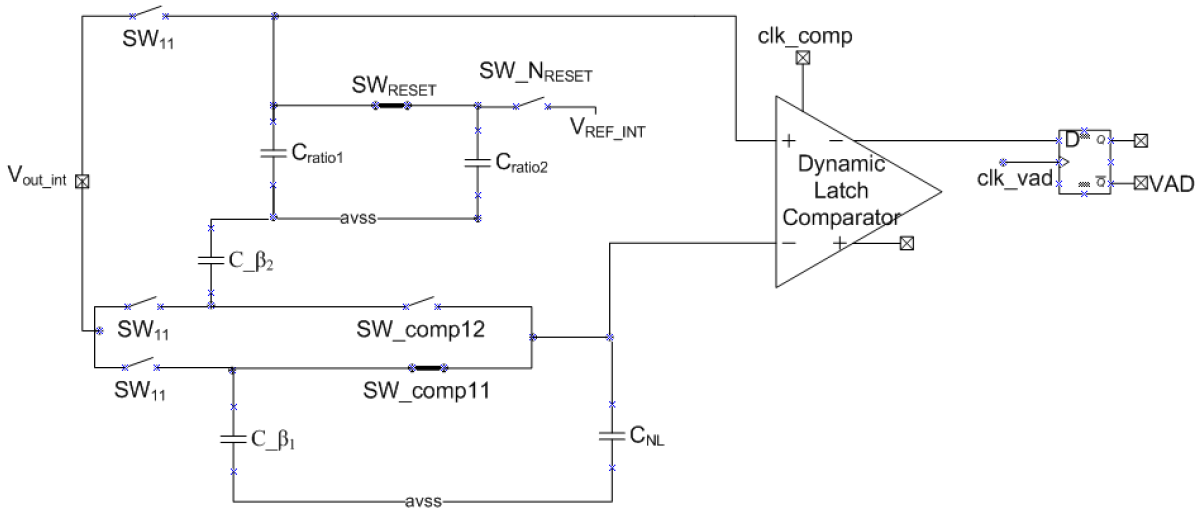


Figure 3.27: Circuit configuration during the second comparison

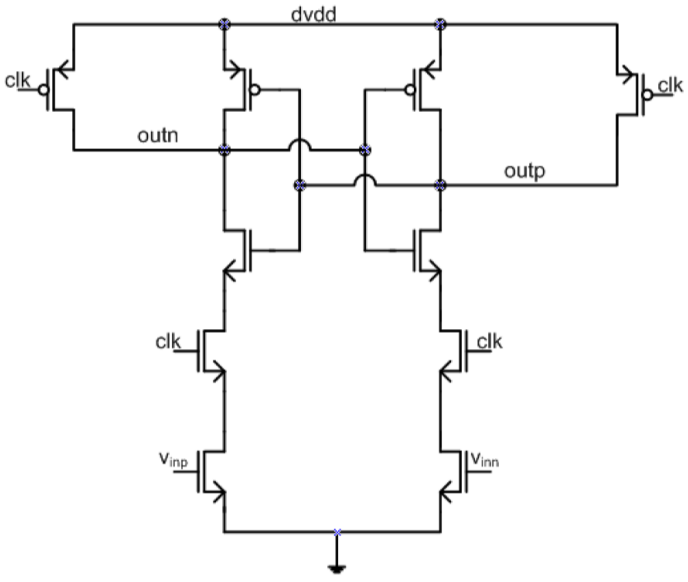


Figure 3.28: Dynamic-latch comparator

At this point, the second comparison takes place (VAD decision) with the circuit configuration shown in Figure 3.27, producing the VAD signal.

The dynamic-latch comparator schematic is shown in Figure 3.28. The input signals V_{inp} and V_{inn} are connected to C_{ratio1} and C_{NL}, respectively. The clock *clk* driving the comparator is *clk_comp* in Figure 3.25. Throughout the integration period and the reset period between the two comparisons *outn* and *outp* are connected to the supply voltage and the comparator is turned off. This solution has been chosen to reduce the power consumption. Indeed, considering the integration period of 16 ms, the average current consumption of this block is of the order of 0.2 nA, leading to a power consumption of 0.36 nW. Figure 3.29 reports the results of 100 montecarlo simulations of the comparator offset, featuring a mean value of 361 μV and a standard deviation of 1 mV.

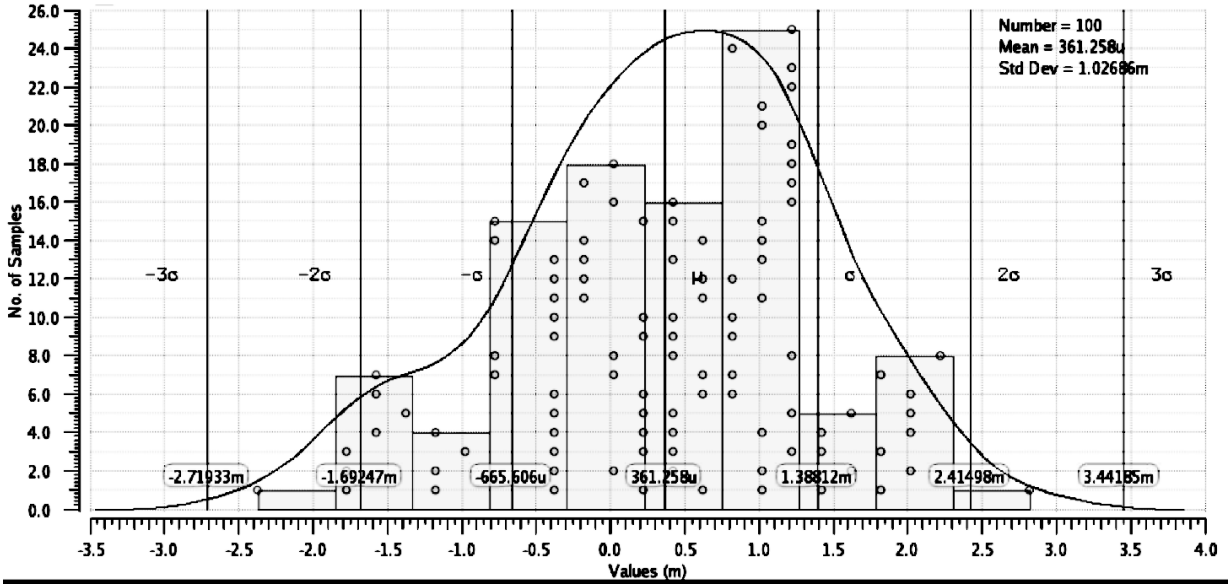


Figure 3.29: Comparator offset over 100 Montecarlo simulations

Chapter 4

Measurement Results

The proposed VAD system has been implemented with a 180-nm CMOS technology. In order to verify the performance achieved by the circuit realized both with continuous-time (CT) and switched (SW) resistors, a test-chip has been fabricated with four versions of the circuit:

- CT resistor with analog buffer;
- CT resistor without analog buffer;
- SW resistor with analog buffer;
- SW resistor without analog buffer.

The circuits with analog buffer have intermediate test points to detect the correct operation of the circuit. In particular, it is possible to verify:

- Integration period clock (SW_{11});
- Comparator clock (clk_comp);
- Noise level decision clock (clk_nl);
- Integration capacitance reset clock (SW_{RESET});
- VAD decision clock (clk_vad).

Figure 4.2 shows a microphotograph of the test chip. The area of the circuit implemented with continuous-time resistors is 0.19 mm^2, whereas the area of the circuit implemented with switched resistors is 0.14 mm^2.

Figure 4.3 shows the measured waveforms of the clock phases obtained from the test-chip, starting from a 3.068-MHz master clock. The decision phase has a pulse width of 100 μs and all the other waveforms are in line with the simulations (Figure 3.25).

The characterization of the VAD circuit has been performed with an audio file 18-minute

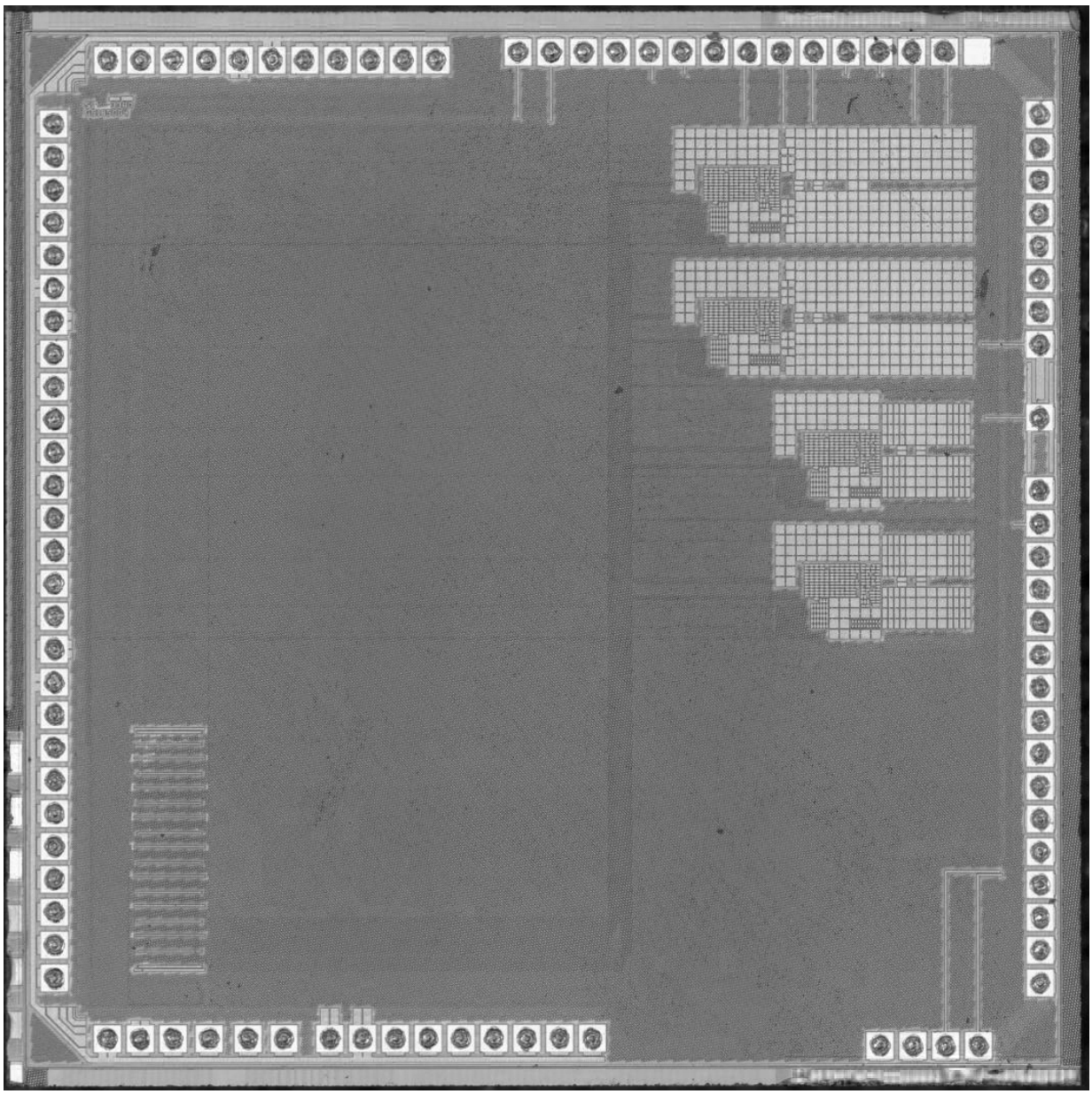


Figure 4.1: Microphotograph of the test chip

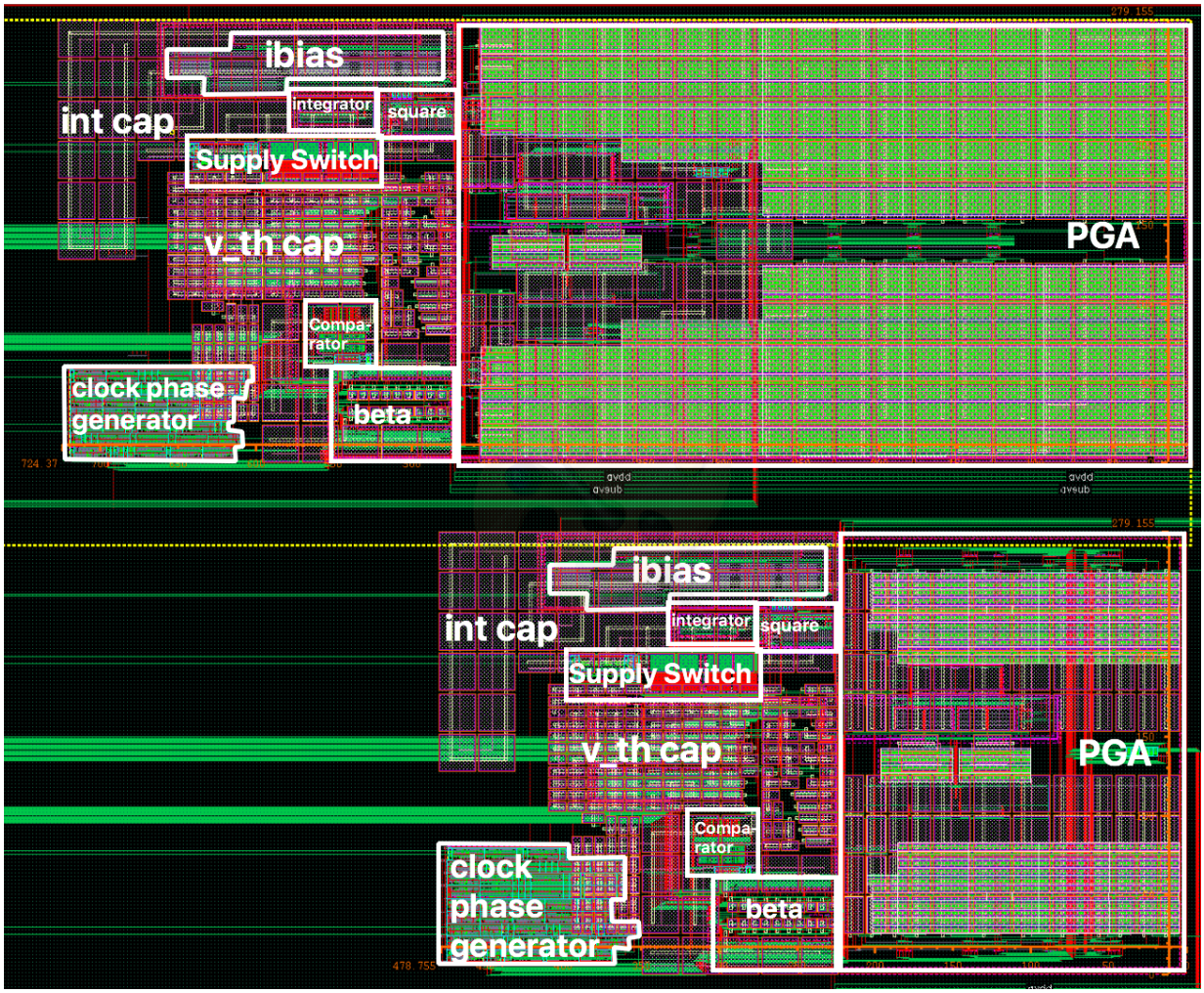


Figure 4.2: Test chip layout detail: CT resistor (above) and SW resistor (bottom)

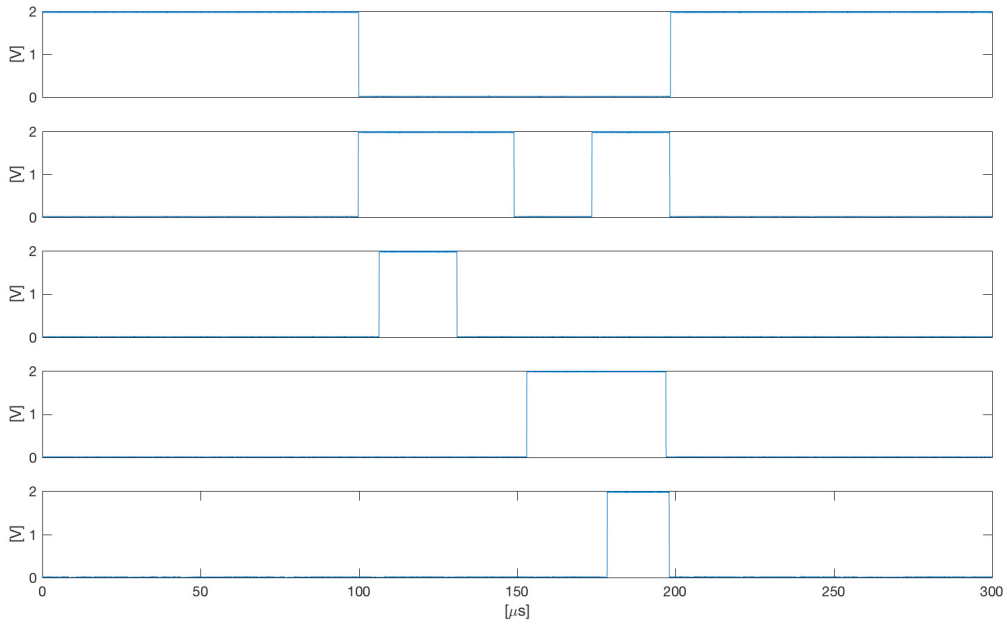


Figure 4.3: Measured clock phases

long (Figure 4.4), that has different types of noise and voice sources. This file has been provided to the circuit through the audio output of a computer. The inputs for the audio signal on the test chip are common for all of the circuit versions. The current reference of 200 nA has been implemented with a 9-MΩ resistance connected between the 1.8-V supply voltage and the drain of a diode-connected NMOS transistor. Each circuit version has a separate input for the current, in order to allow the activation of only one version at a time. The 3.068-MHz master clock is provided through a function generator, while the power-supply voltage and the integrator reference voltage $V_{REF\_INT} = 0.5$ V are provided through an external voltage generator. These are all the analog inputs of the test-chip. In order to write the programming register to select the desired preamplifier gain, integration period, integration capacitance, C_β_1 value, C_β_2 value, and C_{ratio2} value, an I^2C interface has been used.

The configuration of the programmable parameter used for the measurements (unless differently specified) is the following:

- Gain = 12 dB
- Integration period = 16 ms;

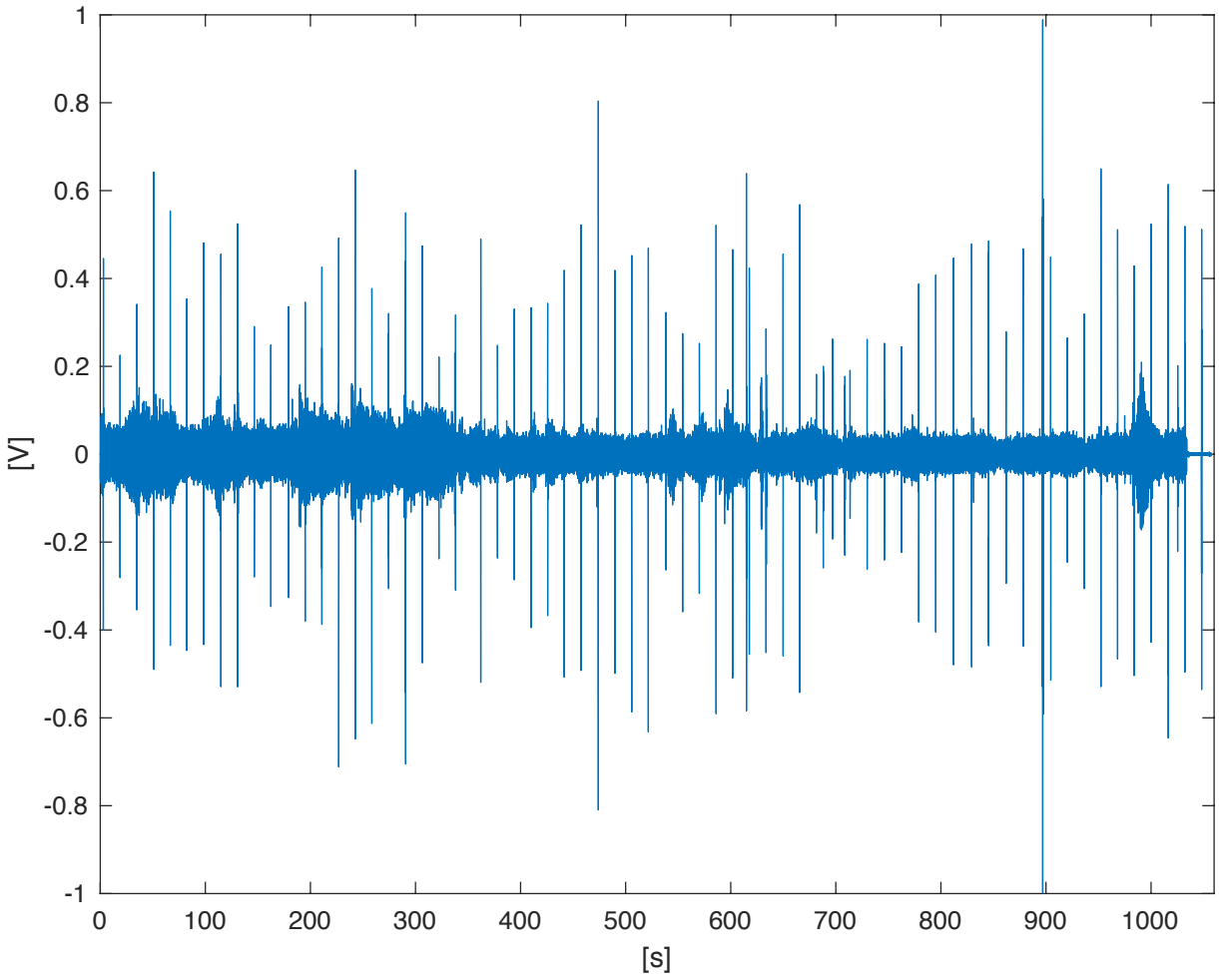


Figure 4.4: Audio frame used for testing

Table 4.1: Measured VAD errors with Gain = 12 dB, Integration Period = 16 ms, Integration Capacitance = 40 pF, $\beta_1 = 0.95$, $\beta_2 = 0.995$, and $th_{SP} = 0.42$

Topology	**Total Errors** [%]	**FP** [%]	**FN** [%]
CT Resistor	0.7037	0.3111	0.3926
SW Resistor	0.7925	0.4681	0.3244
Ideal Model with Noise	0.2615	0.0712	0.1903

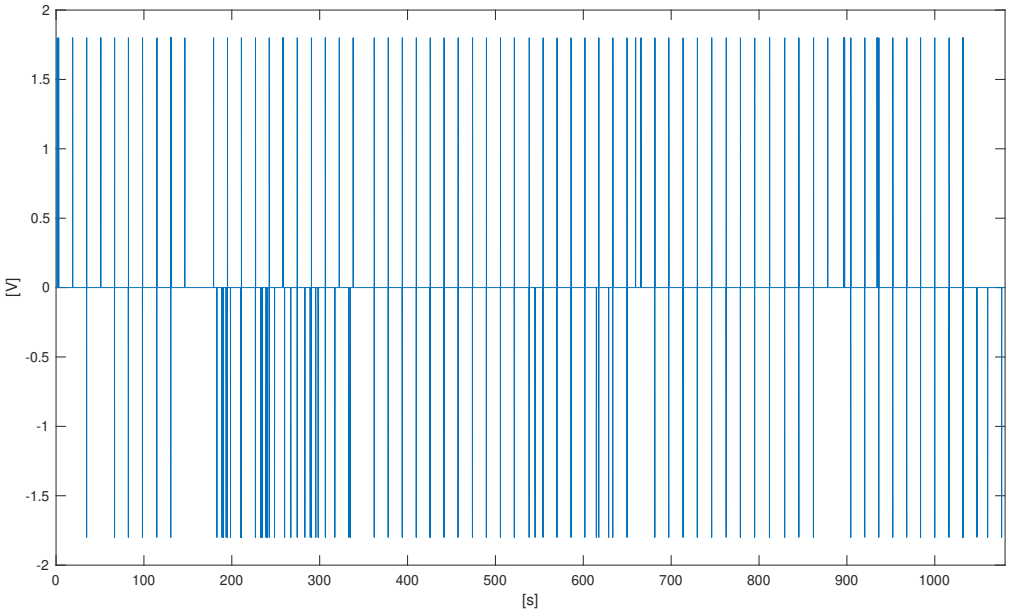
- Integration capacitance = 40 pF;
- $C\_\beta_1 = 0.5$ pF $\Longrightarrow \beta_1 = 0.95$;
- $C\_\beta_2 = 0.05$ pF $\Longrightarrow \beta_1 = 0.995$;
- $C_{ratio2} = 7$ pF $\Longrightarrow th_{SP} = 0.42$.

A comparison has been made between the measured results of the two circuit versions obtained with the input signal of Figure 4.4, which contains 67500 voice events and background noise, and the output of an ideal model, implemented in Matlab, with as input the same audio file. The achieved results are reported in Figure 4.5a for the CT-resistor configuration and in Figure 4.5b for the SW-resistor configuration. Table 4.1 summarizes the errors determined as follows:

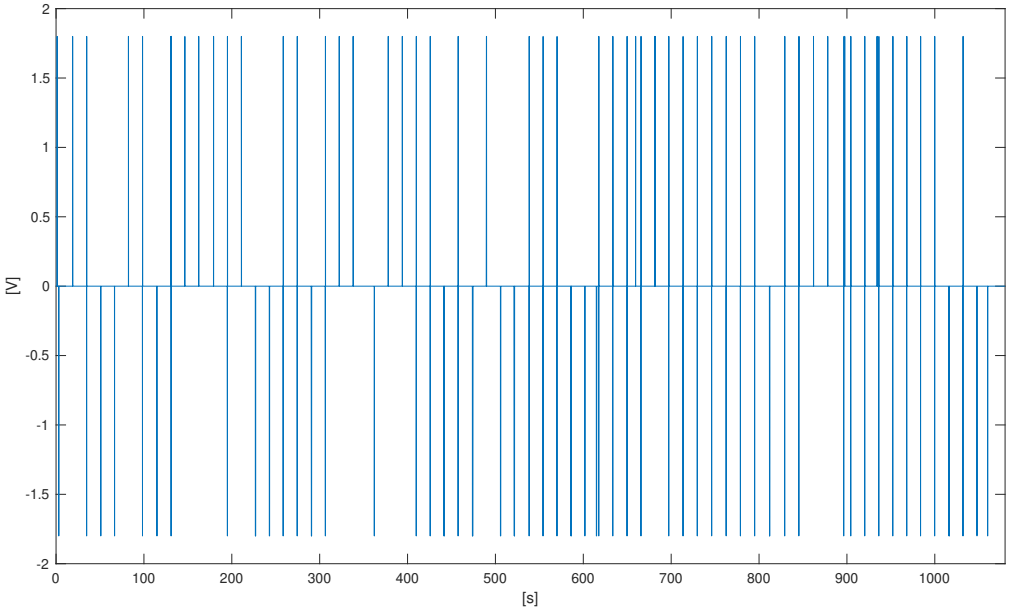
- Total Errors = (# of errors/# of decisions) $\cdot$ 100;
- False Positive (FP) = (# of FP/# of decisions) $\cdot$ 100;
- False Negative (FN) = (# of FN/# of decisions) $\cdot$ 100.

False positive errors occur when there is no voice in the signal and a VAD is made, whereas false negative errors occur when there is voice in the signal and no VAD is made, the total number of errors is the sum of false positive and false negative errors. In Table 4.2 the performances of the proposed circuit are compared with the state-of-the-art.

Finally, Table 4.3, Table 4.4, and Table 4.5 report the measurement results obtained with different setup configurations. The achieved performances are substantially similar, independently of the setup, confirming the robustness of the proposed solution.



(a)



(b)

Figure 4.5: Errors in the VAD signal with respect to the ideal model for the CT-resistor circuit (a) and the SW-resistor circuit (b)

Table 4.2: Comparison with the state-of-the-art

	This Work	**Yang, ISSCC 2018**	**Price, ISSCC 2017**	**Esser, PNAS 2016**	**Badami, ISSCC 2015**	**Raychowdhury, JSSC 2013**
Technology	180 nm	180 nm	65 nm	28 nm	90 nm	32 nm
Input	Mic	Passive Mic	Dig. Sound	Dig. Feature	Passive Mic	Dig. Sound
Feature	Analog - events	Analog - events	Digital	Software	Analog	Digital
Channel #	1	16	32	36	16	32
Freq Range	300 - 6.8k	100 - 5k	N/A	N/A	75 - 5k	11k - 62M
Classifier	Analog Energy Aaveraging	Digital Binarizd D.N.N.	Digital Fixed-Point D.N.N.	Digital Spiking N.N.	Mixed-Signal Decision Tree	Digital Energy-Based Decision Rule
Power [μW]	1.62	1.0	22.3	26100	6	~300
Area [mm^2]	0.14 - 0.19	2.52	2.08	N/A	3	N/A
Rate [s^{-1}]	125-62.5-31.25	100	100	1539	N/A	32600
Classification Dataset	N/A	AURORA4 mixed w/ DEMAND	AURORA2	TIMIT mixed w/ NOISEX	NOISEUS	N/A
Classification Accuracy	99.29% accuracy CT-R 99.21% accuracy SW-R	Speech/non-speech hit-rate 84%/85% 10 dB SNR, restaurant noise	10% EEr 7dB SNR, unspecified context	95.42% accuracy unspecified SNR/context	Speech/non-speech hit rate 89%/85% 12 dB SNR, babble noise	97% accuracy unspecified SNR/context

Table 4.3: Measured VAD errors with Gain = 12 dB, Integration Period = 16 ms, Integration Capacitance = 40 pF, $\beta_1 = 0.99$, $\beta_2 = 0.95$, and $th_{SP} = 0.69$

Topology	**Total Errors [%]**	**FP [%]**	**FN [%]**
CT Resistor	0.4859	0.2563	0.2296
SW Resistor	0.6415	0.4519	0.1896
Ideal Model with Noise	0.2615	0.0712	0.1903

Table 4.4: Measured VAD errors with Gain = 6 dB, Integration Period = 16 ms, Integration Capacitance = 20 pF, $\beta_1 = 0.99$, $\beta_2 = 0.95$, and $th_{SP} = 0.57$

Topology	**Total Errors [%]**	**FP [%]**	**FN [%]**
CT Resistor	0.36	0.0785	0.2815
SW Resistor	0.7141	0.4104	0.3037
Ideal Model with Noise	0.2615	0.0712	0.1903

Table 4.5: Measured VAD errors with Gain = 0 dB, Integration Period = 16 ms, Integration Capacitance = 10 pF, $\beta_1 = 0.99$, $\beta_2 = 0.95$, and $th_{SP} = 0.82$ for the CT-resistor circuit and $th_{SP} = 0.85$ for the SW-resistor circuit

Topology	**Total Errors [%]**	**FP [%]**	**FN [%]**
CT Resistor	0.4607	0.1022	0.3585
SW Resistor	0.7245	0.3793	0.3452
Ideal Model with Noise	0.2615	0.0712	0.1903

Conclusions

In this Thesis a fully-analog voice activity detection system implemented in 180-nm CMOS technology, which achieves a current consumption of 0.9 μA with a supply voltage of 1.8 V, has been presented. The circuit is composed of three main blocks: a preamplifier, a signal energy computation circuit and decision-making circuit. The preamplifier reads-out the signal coming from the microphone with a variable gain ranging from -12 dB to $+12$ dB with steps of 6 dB and a bandpass transfer function with poles at 300 Hz and 6.8 kHz. This block has been designed with continuous-time resistors and with switched resistors. The total area is 0.19 mm^2 for the circuit with the continuous-time resistors and 0.14 mm^2 for the circuit with switched resistors. The second block performs the computation of the signal energy taking advantage of the MOS transistor quadratic current-voltage characteristic to square the signal, which is then integrated over a resettable capacitance. Finally, the third block produces the voice activity detection signal, by comparing the signal energy with the noise level, which is updated after every audio frame. The most significant measurement results are reported in Table 4.6. The proposed fully analog voice activity detection system achieves performance comparable or superior to state-of-the-art digital solutions with a very low power consumption and small area. Moreover, in spite of the analog implementation, the circuit is quite robust against parameter variations.

Table 4.6: Measured results

Topology	**Total Errors** [%]	**FP** [%]	**FN** [%]
CT Resistor	0.7037	0.3111	0.3926
SW resistor	0.7925	0.4681	0.3244
Ideal Model with Noise	0.2615	0.0712	0.1903

Bibliography

[1] D. W. Xiao-Lei Zhang, "Boosting contextual information for deep neural network based voice activity detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 2, pp. 252–264, February 2016.

[2] T. Hughes and K. Mierle, "Recurrent neural networks for voice activity detection," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, October 2013, pp. 7378–7382.

[3] J. G. Michael Price and A. P. Chandrakasan, "A low-power speech recognizer and voice activity detector using deep neural networks," *JSSC*, vol. 53, no. 1, pp. 66–75, Jan. 2018.

[4] B. Y. et al., "Development of robust vad schemes for voice operated switch application in aircrafts," in *2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*. IEEE, July 2016, pp. 191–195.

[5] W. M. Komail M. H. Badami, Steven Lauwereins and M. Verhelst, "A 90 nm cmos, 6 μW power-proportional acoustic sensing frontend for voice activity detection," in *IEEE JSSC*, vol. 51, no. 1, 2016, pp. 291–302.

[6] F. H. M. Niermann and P. Vary, "Speech-codebook based soft voice activity detection," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, August 2015, pp. 4335–4339.

[7] Y. L. Qinghua Huang, Dongmei Wang, "Single channel speech enhancement based on prominent pitch estimation," in *IET International Communication Conference on Wireless Mobile and Computing (CCWMC)*. IET, July 2009, pp. 205–208.

[8] *A Pitch Based VAD Adopting Quasi-Ansi 1/3 Octave Filter Bank with 11.3 ms Latency for Monosyllable Hearing Aids.* IEEE, December 2013.

[9] G. Farahani, "Autocorrelation-based noise subtraction method with smoothing, over-estimation, energy, and cepstral mean and variance normalization for noisy speech recognition," *EURASIP Journal on Audio, Speech, and Music Processing*, pp. 1–16, June 2017.

[10] M. Y. et al., "A 1μW voice activity detector using analog feature extraction and digital deep neural network," in *ISSCC*, March 2018, pp. 346–348.

[11] A. P. C. Michael Price, James Glass, "A scalable speech recognizer with deep-neural-network acoustic models and voice-activated power gating," *ISSCC*, February 2017.

[12] K. B. et al., "Context-aware hierarchical information-sensing in a 6μW 90nm cmos voice activity detector," *ISSCC*, February 2015.

[13] A. R. et Al., "A 2.3 nJ/frame voice activity detector-based audio front-end for context-aware system-on-chip applications in 32-nm cmos," *JSSC*, vol. 48, no. 8, pp. 1963–1969, May 2013.

[14] S. K. E. et al., "Convolutional networks for fast, energy-efficient neuromorphic computing," *PNAS*, vol. 113, no. 41, pp. 11 441–11 446, 2016.

[15] N. W. Thein Htay Zaw, "The combination of spectral entropy, zero crossing rate, short time energy and linear prediction error for voice activity detection," in *20th International Conference of Computer and Information Technology (ICCIT).* IEEE, February 2017, pp. 1–5.

[16] V. M. Horderlin Vrangel Robles Vega and L. Martinez, "Vad algorithms energy-based and spectral-domain applied in river plate castilian," in *XXI Symposium on Signal Processing, Images and Artificial Vision (STSIVA).* November: IEEE, 2016, pp. 1–5.

[17] P. S. V. Nitin N Lokhande, Navnath S Nehe, "Voice activity detection algorithm for speech recognition applications," in *International Conference in Computational Intelligence (ICCIA)*, 2011, pp. 5–7.

[18] A. S. et al., "Vad techniques for real-time speech transmission on the internet," in *First Asian Himalayas International Conference on Internet*. IEEE, November 2002, pp. 46–50.

[19] X. Wang and L. Qu, "The self-adaptive voice activity detection algorithm based on time- frequency parameters," *The Open Automation and Control Systems Journal*, no. 6, pp. 1661–1668, December 2014.

[20] S. B. /jebara, "Multi-band coherence features for voiced-voiceless-silence speech classification," in *2nd International Conference on Information & Communication Technologies*. IEEE, October 2006, pp. 1248–1253.

[21] S. R. M. P. Sarfaraz Jelil, Rohan Kumar Das and R. Sinha, "Role of voice activity detection methods for the speakers in the wild challenge," in *NCC*, October 2017, pp. 1–6.

[22] S. A. S.A. Soleimani, "Voice activity detection based on combination of multiple features using linear/kernel discriminant analyses," in *3rd International Conference on Information and Communication*. IEEE, May 2008, pp. 1–5.

[23] D. R. B. Tejus Adiga M, "Improving single frequency filtering based voice activity detection (vad) using spectral subtraction based noise cancellation," in *International conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*. IEEE, June 2016, pp. 18–23.

[24] G. Aneeja and B. Yegnanarayana, "Single frequency filtering approach for discriminating speech and nonspeech," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 4, pp. 705–717, February 2015.

[25] M. C. et Al., "Cap-less audio preamplifiers for silicon microphones," in *IEEE SENSORS*. IEEE, October 2016, pp. 1–3.

[26] M. C. et al., "Mems microphone fully-integrated cmos cap-less preamplifiers," in *PRIME*. IEEE, June 2017, pp. 37–40.