CHARGE-BASED ANALOG CIRCUITS FOR RECONFIGURABLE SMART SENSORY SYSTEMS

A Dissertation
Presented to
The Academic Faculty

By

Sheng-Yu Peng

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
in
Electrical and Computer Engineering

School of Electrical and Computer Engineering
Georgia Institute of Technology
August 2008

CHARGE-BASED ANALOG CIRCUITS FOR RECONFIGURABLE SMART SENSORY SYSTEMS

Approved by:

Dr. Paul E. Hasler, Advisor
Professor, School of ECE
Georgia Institute of Technology
Atlanta, GA

Dr. Maysam Ghovanloo
Professor, School of ECE
Georgia Institute of Technology
Atlanta, GA

Dr. David V. Anderson
Professor, School of ECE
Georgia Institute of Technology
Atlanta, GA

Dr. Bradley A. Minch
Professor, Department of ECE
Franklin W. Olin College of Engineering
Needham, MA

Dr. F. L. Degertekin
Professor, School of ECE
Georgia Institute of Technology
Atlanta, GA

Date Approved: June 2008

DEDICATION

To my loving family ...

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to everyone who has supported and helped me during my long journey of pursuing Ph.D. degree. Without them, I would not have made it this far.

First of all, I would like to thank my advisor, Dr. Paul Hasler, for giving me all the opportunities at Georgia Tech in all aspects along with his invaluable guidance, constant support, and persistent encouragement, which make this research and dissertation possible. I am especially grateful for his kindness and amicability. To me, he is not only an advisor during my academic years but also as a friend in my life as well as on the basketball court.

I would also like to thank Dr. Bradley Minch for his inspiring lectures and unreserved sharing that bring me into the field of analog signal processing. I am also thankful for his consistent advice and help since the beginning of my journey.

I would also like to extend my appreciation to Dr. David Anderson and Dr. F. L. Degertekin for the insightful comments on my research and all the advice on my academic studies and help in my career planning. I would like to thank Dr. Jennifer Michaels for serving on my dissertation proposal committee, and Dr. Maysam Ghovanloo for serving on my dissertation defense committee.

I would like to say thank you to Kofi Odame and Walter Huang for their help, patience, and companionship. Many thanks to members of CADSP research group for helping and teaching me lots of knowledge and skills that I would never have learned from class. I am very thankful to Ethan, Christal, David, Erhan, Gail, Venkatesh, Christopher, Guillermo, Venkatesh, David, Ryan, Degs, Huseyin, and Arindam, Shakeel, Gokce, Chris, Jordan, Scott, Shubha, Craig, Richie, Stephen, Csaba, Michael, Sourabh, and Shyam for their friendship and support. I will also like to thank collaboration partners working in the Love Building. I benefit a lot from working with Dr. Mustafa Karaman, Dr. Mujdat Balantekin, Dr. Rasim Guldiken and Jaime Zahorian. I would also like to thank Yu Tsao for his

help in this research. He and lots of friends in TSA make my stay at Atlanta a memorable and enjoyable experience.

Special thanks go to Fanpei Yang for her listening, encouragement, and love, which have brought me the strength to strive through all the harsh times without feeling alone during my graduate life. Finally, I am deeply indebted to my parents and my brother for all the sacrifices they have made for me to succeed throughout my life and for their unconditional support and endless love.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Sensory microsystems have attracted a tremendous amount of research interest because of their potential impact on a wide variety of applications. These systems receive signals through sensors and perform local signal processing to reduce the information bandwidth for subsequent transmission and conversion or to interact with the environment without explicit users' involvement. These miniaturized systems can be used in medical prosthetics or personal mobile and entertainment devices that are expected to bring significant improvement in the quality of our daily lives.

In the conventional analog-digital partitioning scheme, the power consumption has become a bottleneck factor in determining both the functionality and the feasibility of these microsystems. Although analog signal processing approaches have shown superior power efficiency, the traditional trimming techniques used to compensate for device mismatches usually consume lots of power and chip area. Besides, lack of programmability also makes analog signal processing approaches less attractive.

The ability to control the charge at floating nodes overcomes the shortcomings of conventional analog approaches and provides an additional design degree of freedom. The notion of designing circuits based on charge sensing, charge adaptation, and charge programming is explored in this research. This design concept leads to a low-power capacitive sensing interface circuit that has been designed and tested with a MEMS microphone and a capacitive micromachined ultrasonic transducer. Moreover, by using the charge programming technique, a designed floating-gate based large-scale field-programmable analog array (FPAA) containing a universal sensor interface sets the stage for reconfigurable smart sensory systems. Based on the same charge programming technique, a compact programmable analog radial-basis-function (RBF) based classifier and a resultant analog vector quantizer have been developed and tested. Measurement results have shown that the analog RBF-based classifier is at least two orders of magnitude more power-efficient

than an equivalent digital processor. Furthermore, an adaptive bump circuit that can facilitate unsupervised learning in the analog domain has also been designed. A projection neural network for a support vector machine, a powerful and more complicated binary classification algorithm, has been proposed. This neural network is suitable for analog VLSI implementation and has been simulated and verified on the transistor level. These analog classifiers can be integrated at the interface to build smart sensors.

This work contributes to the reconfigurable smart sensory systems by exploring the technologies of capacitive sensing, reconfigurable analog array with universal sensor interface, and programmable analog classifiers. The results from this research have laid down the foundations for developing highly power-efficient reconfigurable smart sensory systems.

CHAPTER 1

RECONFIGURABLE SMART SENSORY SYSTEM

The integration of sensors, interface circuits, and signal processing circuitry has given rise to the developments of a great diversity of sensory microsystems that are expected to bring positive and revolutionary impacts on our daily lives. This chapter describes the challenges in developing these sensory microsystems and explains how reconfigurable smart sensory systems address these issues based on the floating-gate technologies.

1.1 Advanced Sensory Microsystems

The advancements in silicon technologies not only have made electronic devices smaller and faster but also have enabled the fabrication of a great diversity of micromachined transducers. Today, a large number of sensors can be integrated with electronics to create a variety of innovative sensory microsystems. Such microsystems receive external signals through sensors and then perform local signal processing to refine the information for subsequent processing or to interact with the environment directly without explicit users' involvement. The block diagram of a typical sensory system is shown in Figure 1.1.

A large number of external signals received from the sensors and the tremendous computational capabilities provided by the integrated electronics give designers great opportunities to develop advanced "smart" sensory microsystems with features that cannot be

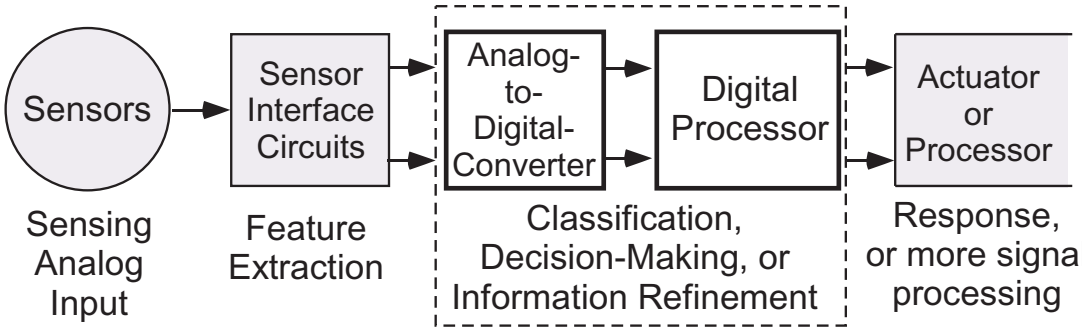


Figure 1.1. The block diagram of a typical sensory system.

achieved before. At the same time, system designers also have to face severer power constraints than ever. The available power resources in microsystems are limited because of the small form factor. Low-power operation can prolong the life time of the system or can decrease the frequency for recharge. Even in some cases that the power is supplied externally and is delivered through wires or inductive coils, the power dissipation can overheat the microsystems due to the high power density. These more stringent power constraints determine both the functionality and the feasibility of microsystems.

1.2 Analog Signal Processing

Analog signal processing has shown as a promising approach to address the power issues because of its superior efficiency. From the projected trend in Figure 1.2, the power savings of the analog signal processing is equivalent to a 20 year leap in digital technology. Furthermore, if the meaningful information can be refined using analog signal processing techniques at the sensor interface, the bandwidth for subsequent transmission, conversion, and processing can be much reduced. In some cases, the analog-to-digital conversion can be avoided altogether. As a result, the system can achieve much better efficiency and can overcome the difficulties associated with the power constraints.

Although analog signal processing plays a significant role in addressing the power challenges of advanced sensory microsystems, analog design itself is difficult. Since analog circuit performance is vulnerable to fabrication imperfections, the means to compensate for device mismatches and circuit offsets is necessary. In conventional analog approaches, these techniques consume lots of power and silicon area, which wipes out many advantages of analog signal processing. Besides, lack of programmability and reconfigurability is another shortcoming of conventional analog systems. This is also one of the major reasons that analog signal processing is not as popular as digital signal processing.

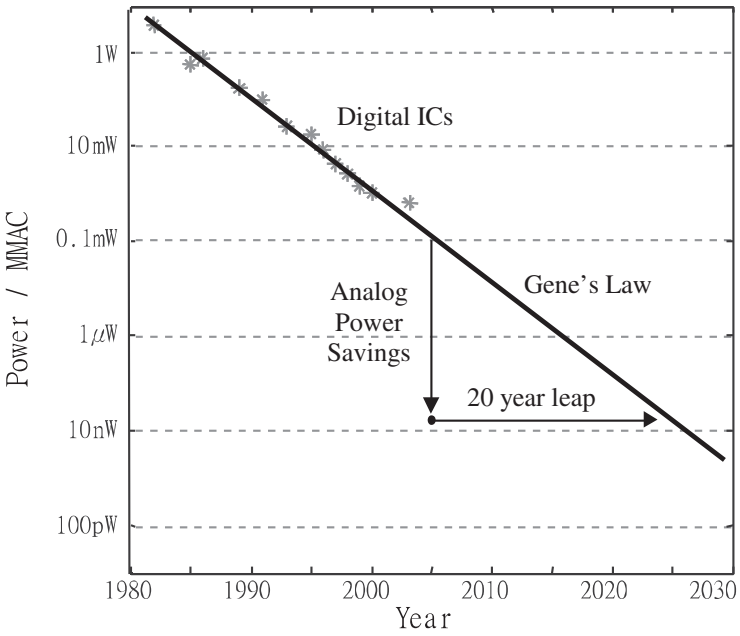


Figure 1.2. Power efficiency comparisons between analog and digital signal processing [1].

1.3 Charge-Based Analog Design

Floating-gate technology has been exploited to get over the shortcomings of conventional analog approaches. Floating-gate techniques enable precise charge programming on floating nodes and provide an additional degree of freedom in designing analog circuits and systems. The new freedom allows designers to make use of capacitive circuits where floating nodes exist. It also prompts the notion of charge-based analog design based on techniques of charge sensing, charge adaptation, and charge programming.

This research explores the notion of charge-based analog circuits that facilitate reconfigurable smart sensory microsystems. Inspired by a capacitive feedback circuit, a new capacitive sensing interface circuit that senses the charge variation ascribed to a varying capacitor has been designed. The output DC level of this circuit is adjusted using charge adaptation techniques. This capacitive sensing circuit can achieve low-power consumption and has a large dynamic range. The charge programming technique can be utilized to compensate for device mismatches and circuit offsets without consuming extra power. It also provides an effective and efficient way to implement analog memories. Therefore,

floating-gate based analog circuits can perform highly power-efficient programmable signal processing in an advanced reconfigurable smart sensory microsystem.

1.4 Reconfigurable Smart Sensory Chip

Thanks to the emerging MEMS technologies, a variety of sensors has been developed and the demands for designing their interface circuits have grown rapidly. A reconfigurable smart sensory chip would be a useful tool for fast prototyping innovative ideas of sensory microsystems. It can greatly shorten design-testing cycles and hence reduce the costs of system development.

The block diagram of a reconfigurable smart sensory chip is shown in Figure 1.3. The chip can interface with different kinds of sensors and can perform different analog algorithms for different applications. Depending on the types of sensors employed in the microsystem, appropriate interface circuits are synthesized in the universal sensor interface (USI) blocks to transduce signals received from the outside world. Components inside the configurable analog blocks (CABs) are configured to perform feature extraction or other analog signal processing tasks. Resultant features can be fed to a programmable analog classifier for further information refinement. The reconfigurability of the chip is achieved by the interconnect network that connects component terminals to form different circuit topologies.

This chip offers sensor specialists a fast way to test their sensors and provides signal processing experts a programmable platform to perform different power-efficient analog algorithms. Because the output signals are no longer raw data but refined meaningful information, the required processing burdens of subsequent stages can be greatly lessened and the entire system can be very power efficient. This research focuses on the developments of interface circuits and analog classifiers that are critical building blocks of a reconfigurable smart sensory system.

Floating-gate transistors are versatile in the reconfigurable smart sensory chip. They

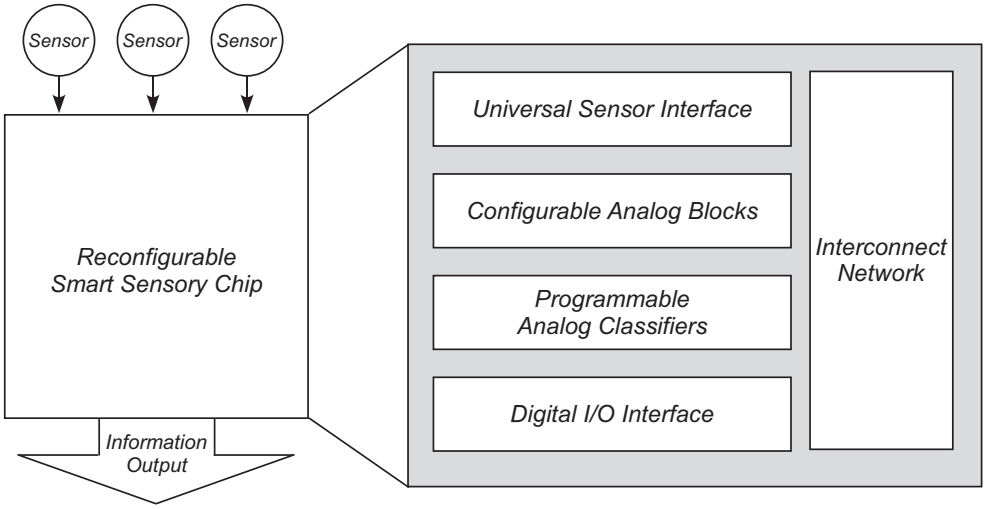


Figure 1.3. The block diagram of a reconfigurable smart sensory chip.

are utilized as compact switches in the interconnect network, as programmable elements in USI and CABs, and as analog memories in analog classifiers. Therefore, this dissertation begins with Chapter 2 presenting some capacitive circuits and an overview of floating-gate technologies. Chapter 3 describes a detail analysis, measurement results, comparisons, and a design procedure of a proposed low-power capacitive feedback charge amplifier for capacitive sensing. This approach has been applied to audio and to ultrasonic applications. The ultrasonic results are discussed in Chapter 4. In Chapter 5, a large-scale field-programmable analog array (FPAA) with integrated universal sensor interface blocks are presented. In Chapter 6, a fundamental analog circuit used in analog classifiers for similarity measure is introduced. This floating-gate circuit has been used to implement a compact and power-efficient analog radial-basis-function-based classifier that is presented in Chapter 7. An adaptive vector quantizer that can perform unsupervised learning is described in Chapter 8. In Chapter 9, a projection neural network for a more complicated binary classifier, the support vector machine, is presented. This projection neural network can be implemented in analog circuits without using any resistors or operational amplifiers and is suitable for large-scale implementations. The main contributions and key milestones that have been achieved in this work are summarized in Chapter 10.

CHAPTER 2

PROGRAMMABLE ANALOG TECHNOLOGIES

Analog signal processing is an attractive approach to developing microsystems because of its superior power efficiency. However, it is not trivial to design analog processors performing accurate computation in the presence of device mismatches. Traditional design techniques used to compensate for fabrication imperfections usually consume tremendous amounts of power and die area, which annuls many of the benefits of analog signal processing. Lack of programmability and reconfigurability is another drawback of conventional analog systems. Recently, floating-gate technologies have been shown as powerful tools to circumvent these shortcomings. Floating-gate transistors have been successfully utilized to trim out amplifier offsets [2], to build data converters [3], to generate programmable references [4], and to make large-scale programmable analog signal processing systems possible [1, 5].

This chapter starts with introducing some capacitive circuits that are associated with floating-gate transistors as well as a capacitive sensing interface circuit proposed in this research. An overview of floating-gate transistors including the structure and the characteristics is then presented. Techniques to precisely program an array of floating-gate transistors are also detailed in this chapter.

2.1 Capacitive Circuits

The capacitive circuit approach is a practical and efficient technique for integrated circuit designers because capacitors are natural elements in a CMOS process. Unlike resistors, capacitors neither dissipate DC power nor do they contribute thermal noise. However, in conventional circuit design, capacitive circuits that include some floating nodes are usually avoided because the charges on the floating nodes are neither predictable nor controllable. Recently, charges on the floating nodes can be precisely programmed using floating-gate

technologies. As a result, capacitive circuits that can effectively calculate the weighted summation of voltages are no longer prohibitive.

If the initial conditions of floating-node charges are not given, Kirchhoff's current and voltage laws are not enough to analyze capacitive circuits. A capacitive circuit needs to be described using the principle of charge conservation. A capacitive voltage divider is shown in Figure 2.1a. If Kirchhoff's current law is used to derive the output voltage, the equality will be given as

$$sC_1(V_{\text{in}} - V_{\text{out}}) = sC_2V_{\text{out}}. \tag{2.1}$$

Apparently the DC output voltage cannot be determined by (2.1) because $s = j\omega = 0$ at DC. If V_{out} is a floating node and the charge conservation principle is applied to that node, the resultant equality can be expressed as

$$Q = C_1(V_{\text{out}} - V_{\text{in}}) + C_2V_{\text{out}}. \tag{2.2}$$

The output voltage can then be derived as

$$V_{\text{out}} = \frac{C_1V_{\text{in}} + Q}{C_1 + C_2} \tag{2.3}$$

$$= \frac{C_1V_{\text{in}}}{C_1 + C_2} + V_{\text{Q}}. \tag{2.4}$$

The output voltage expression, (2.4), of Figure 2.1a is similar to that of a resistive voltage divider. The difference is the extra voltage term, $V_{\text{Q}} = Q/(C_1 + C_2)$, set by the charge at the output floating node.

A more general capacitive circuit with multiple inputs is illustrated in Figure 2.1b. The charge conservation principle can also be applied and the equality is given as

$$Q = C_1(V_{\text{fg}} - V_1) + C_2(V_{\text{fg}} - V_2) + ... + C_n(V_{\text{fg}} - V_n). \tag{2.5}$$

The resultant expression of the floating-node voltage is

$$V_{\text{fg}} = \frac{C_1V_1 + C_2V_2 + ... + C_nV_n + Q}{C_1 + C_2 + ... + C_n} \tag{2.6}$$

$$= \frac{C_1V_1 + C_2V_2 + ... + C_nV_n}{C_{\text{T}}} + V_{\text{Q}}. \tag{2.7}$$

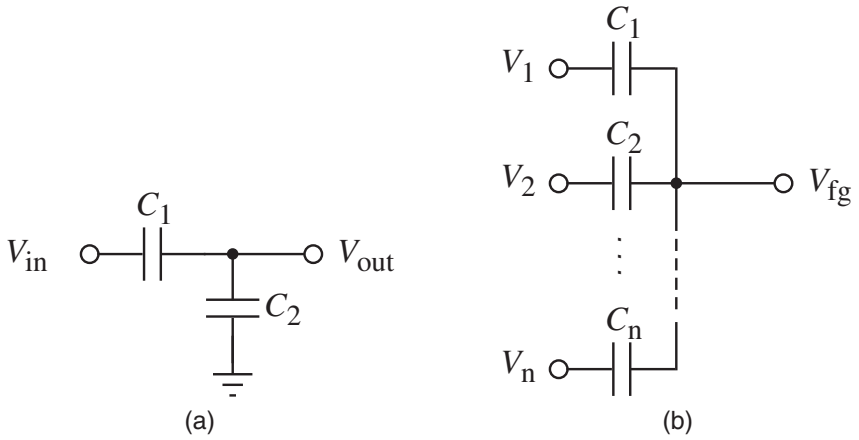


Figure 2.1. The schematics of two capacitive circuits.
(a) A capacitive voltage divider
(b) A capacitive voltage summation

From (2.7), the operation of weighted voltage summation can be easily implemented in Figure 2.1b without consuming DC power.

The capacitive circuit can also be used to form a feedback around an amplifier, as shown in Figure 2.2a. Again, the equality of the charge conservation on the floating node can be expressed as

$$Q = C_1(0 - V_{in}) + C_2(0 - V_{out}), \tag{2.8}$$

and the output voltage can be expressed as

$$V_{out} = -\frac{C_1 V_{in} + Q}{C_2} \tag{2.9}$$

$$= -(\frac{C_1}{C_2} V_{in} + V_Q). \tag{2.10}$$

The closed-loop gain is $-C_1/C_2$, as expected. There is also a charge dependent term, $V_Q = Q/C_2$, in the output voltage expression.

Traditionally, floating-node approaches have been avoided by circuit designers because the charge on a floating node is neither predictable nor controllable. With recent advancements in floating-gate techniques, charges on floating nodes now can be adjusted or programmed [6, 7] and, as a result, capacitive circuits are no longer prohibitive. Stemming

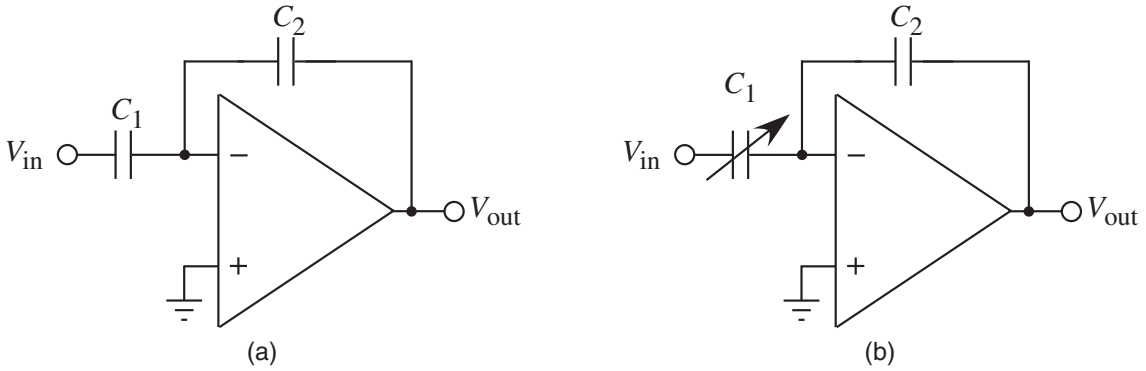


Figure 2.2. The schematics of two capacitive feedback circuits.
(a) A capacitive voltage amplifier
(b) A capacitive sensing charge amplifier

from the circuit in Figure 2.2a, the capacitive circuit in Figure 2.2b has been proposed as a low-power capacitive sensing circuit that will be detailed in the subsequent chapter.

2.2 Floating-Gate Transistor

A floating-gate transistor is a normal transistor with the gate that has no DC path to a fixed voltage. One or multiple external voltages are capacitively coupled into the floating gate. The layout and the circuit schematic of a pFET floating-gate transistor with two input capacitances are shown in Figure 2.3a. The floating gate is made up of the poly1 layer and is surrounded by high-quality silicon-oxide. Under normal operating conditions, the charge stored on the floating gate cannot escape and, therefore, the floating-gate transistor can be used as a non-volatile memory device. The external voltages are applied to the poly2 layers and are capacitively coupled into the floating gate via poly-poly capacitances. If the parasitic capacitance from the n-well to the poly gate is neglected, the floating-gate voltage in Figure 2.3a can be expressed as

$$V_{fg} = \frac{C_1 V_{in1} + C_2 V_{in2} + C_{tun} V_{tun} + Q}{C_1 + C_2 + C_{tun}}. \tag{2.11}$$

Charge on the floating-gate can be adjusted using hot-electron injection and Fowler-Nordheim tunneling. The tunneling mechanism takes place at a tunneling capacitor that is made from poly-silicon and a separate n-well.

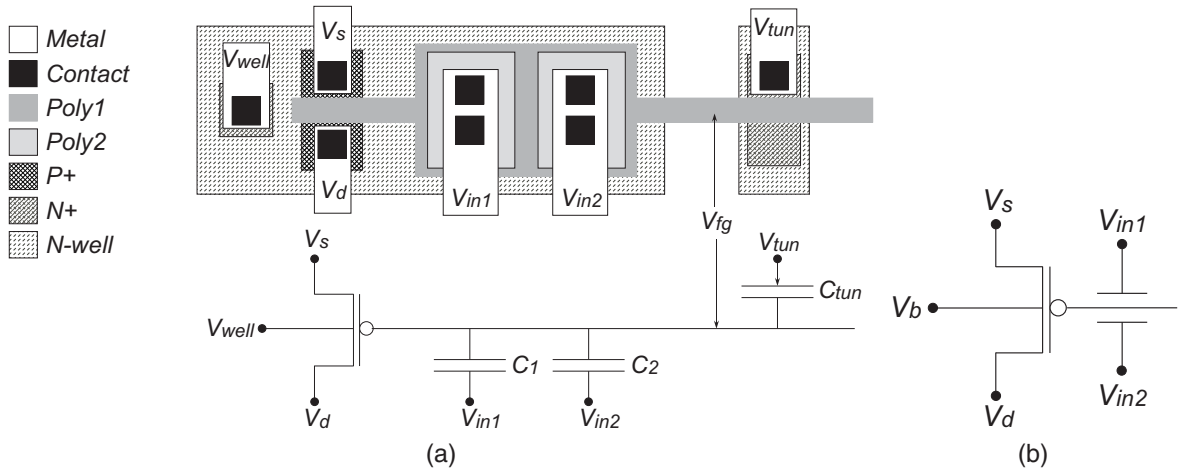


Figure 2.3. The circuit schematic, layout, and symbol of a pFET floating-gate transistor with two input capacitances.
(a) The circuit schematic and the layout
(b) The symbol

From (2.11), a single floating-gate transistor performs the operation of weighted summation of voltages and also functions as an analog memory. Therefore, it is a versatile computation and storage device for analog signal processing. If the input coupling capacitances in Figure 2.3a are matched and the tunneling capacitance as well as other parasitic capacitances is negligible, (2.11) can be approximated as

$$V_{\mathrm{fg}} \approx \frac{1}{2}V_{\mathrm{in1}} + \frac{1}{2}V_{\mathrm{in2}} + V_{\mathrm{Q}}, \tag{2.12}$$

where $V_{\mathrm{Q}} = Q/C_{\mathrm{T}}$ and C_{T} is the total capacitance seen from the floating gate. This two-input floating-gate transistor, the symbol of which is shown in Figure 2.3b, has been utilized in developing compact analog classifiers in this research.

2.3 Floating-Gate Programming

The charge stored on a floating gate is insulated by surrounding oxide. To move the charge across the oxide energy barrier, Fowler-Nordheim tunneling and channel hot electron injection mechanisms are used.

Tunneling arises from the fact that an electron wave function has finite extent. If the energy barrier is thin enough, the extent is sufficient for an electron to penetrate the barrier.

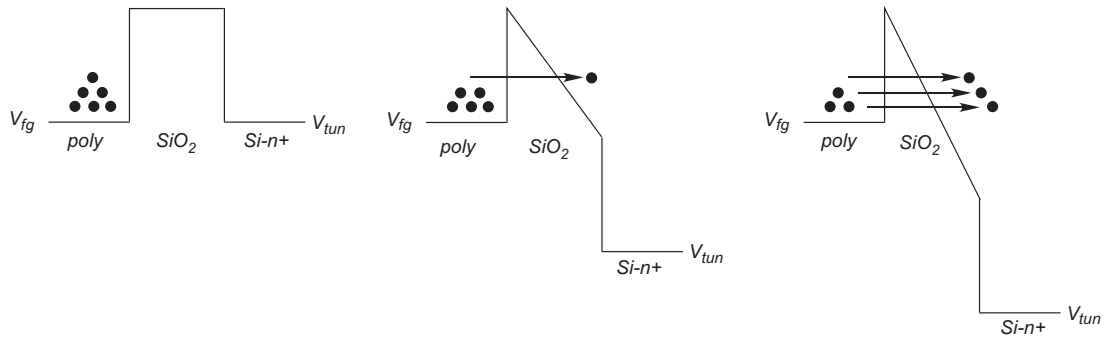


Figure 2.4. The conduction band diagram for tunneling process.

In a tunneling capacitor, the energy barrier of the oxide prevents electrons on the floating gate from jumping to the n-well. If the tunneling voltage is raised, a high electric field across the tunneling capacitor results in decrease of effective thickness of the barrier, which increases the probability of electrons crossing the oxide, as shown in Figure 2.4. In this manner, Fowler-Nordheim tunneling removes electrons from the floating gates through the tunneling capacitor. The tunneling current can be modeled [8] by

$$I_{\text{tun}} = I_0 e^{-\frac{t_{\text{ox}}\mathcal{E}_0}{V_{\text{tun}}-V_{\text{fg}}}}, \tag{2.13}$$

where $\mathcal{E}_0$ is a device parameter. Because of poor selectivity, tunneling currents are used as a global erase. The tunneling voltage used for floating-gate transistors fabricated in a 0.5 μm process is above 15 V.

Channel hot electron injection can transfer electrons from the channel of a MOS transistor to the floating gate. There are two necessary conditions for hot electron injection: sufficient amount of channel current and a high electric field across the channel and the drain terminal. The injection process is illustrated in Figure 2.5. When holes in a pFET transistor travel from the source to the drain, they will be accelerated in the channel-drain depletion region. If the electric field between the channel and the drain is large enough, the holes will cause impact ionization and will generate hot hole-electron pairs. The energy of these hot electrons can be so high that some hot electrons can jump cross the oxide barrier onto the floating gate. Other electrons will flow to the bulk and become bulk current. The

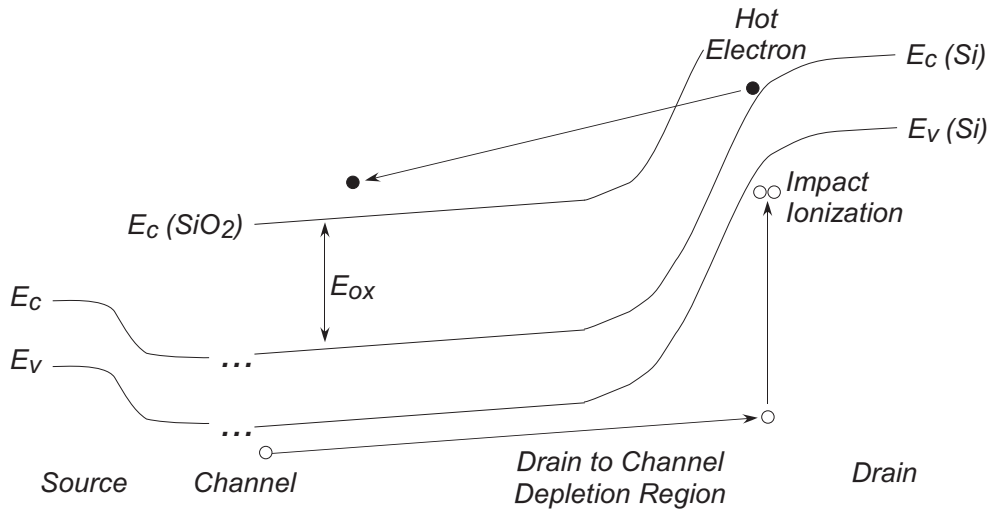


Figure 2.5. The band diagram for channel hot electron injection process.

hot-electron injection current can be modeled [7, 8] as

$$I_{\text{inj}} = I_{\text{inj0}} \left(\frac{I_{\text{s}}}{I_{\text{s0}}} \right)^{\alpha} e^{-\Delta V_{\text{ds}}/V_{\text{inj}}}, \tag{2.14}$$

where I_{s} is the channel current, V_{inj} is a device and bias dependent parameter, and α is very close to 1.

To enable the injection, the voltage between the source and drain needs to be larger than the nominal V_{DD} so that there exists sufficient channel-drain voltage drop for impact ionization. To prevent accidental injection, all terminals of the floating-gate transistor are raised to the injection V_{DD}, which is about 6.5 V for a 0.5 μm CMOS process, during the ramp-up phase. To activate precise injection, the input gate voltage is adjusted to provide enough channel current and a short voltage pulse is applied to the drain. The injection takes place in this pulsing phase. Usually, the pulse width is fixed and the pulse height can be predicted algorithmically. After the injection, all terminal voltages are decreased by the same amount and the source voltage is back to the nominal V_{DD}. A time diagram of the injection process is shown in Figure 2.6. Measured floating-gate characteristics after tunneling and injection are shown in Figure 2.7. Programming the charge on the floating gate is equivalent to adjusting the threshold voltage of a transistor. Therefore, precise

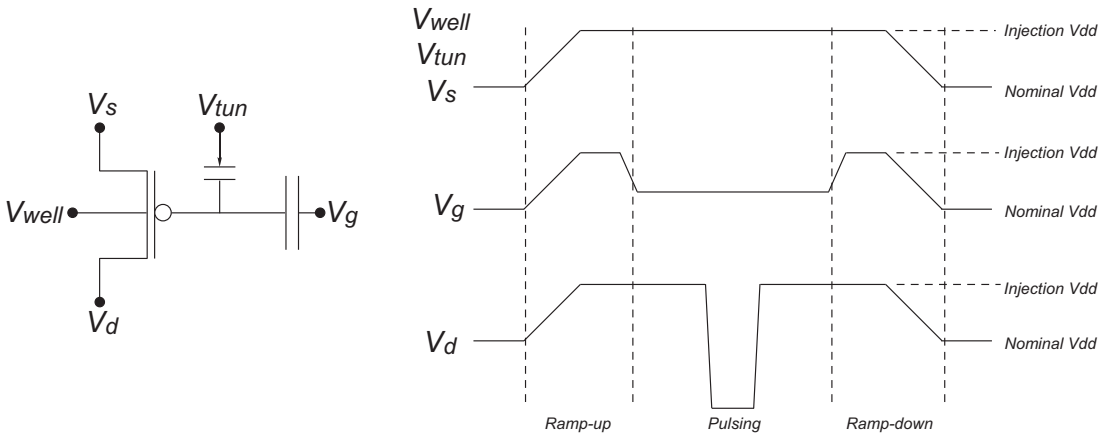


Figure 2.6. The time diagram of injection process.

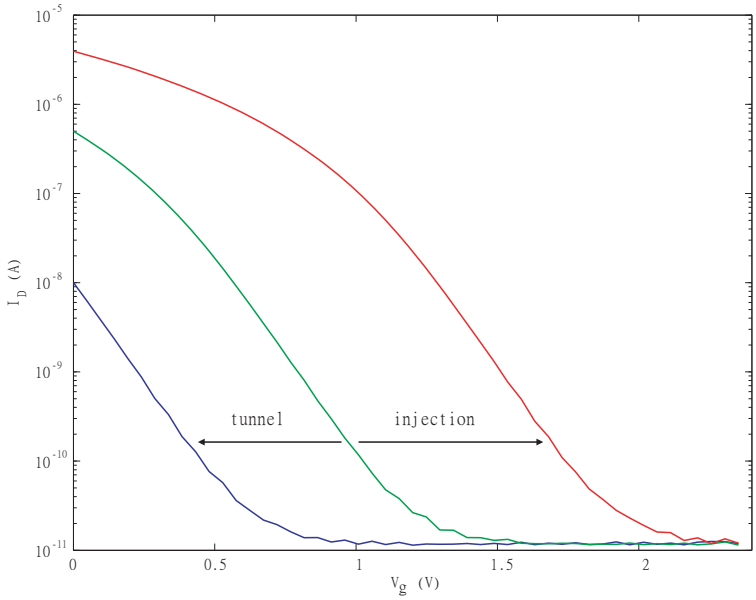


Figure 2.7. The measured I-V curves after tunneling and injection [1].

floating-gate programming can be an effective way to compensate for threshold voltage mismatches.

To predict the required pulse depth for precise programming, an empirical model for injection proposed in [9] is used. Given a short pulse of $V_{\rm ds}$ across a floating-gate device, the injection current is proportional to $\Delta I_{\rm s}/I_{\rm s0}$, where $\Delta I_{\rm s} = I_{\rm s} - I_{\rm s0}$ is the increment of the channel current. From (2.14), the logarithm of this ratio should be a linear function of $V_{\rm ds}$ and a nonlinear function of $\log(I_{\rm s0}/I_{\rm u})$, where $I_{\rm u}$ is an arbitrary unity current. It can be

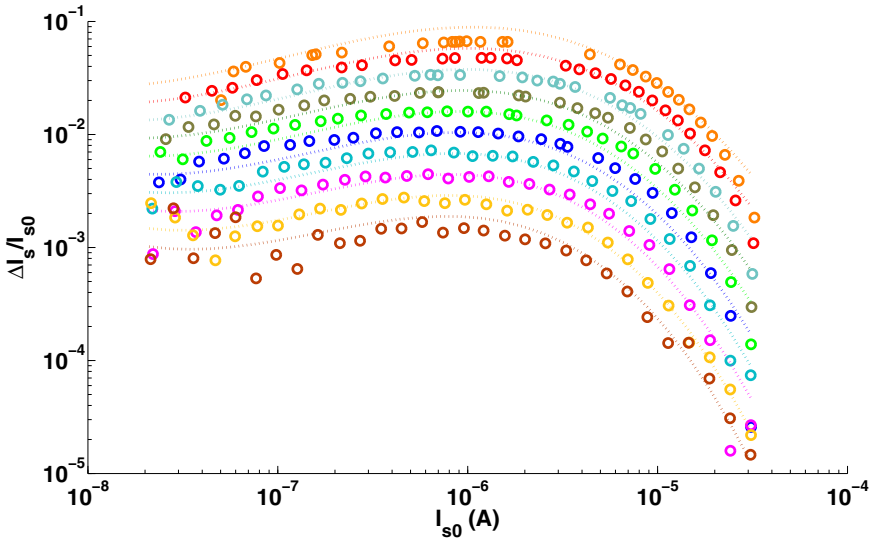


Figure 2.8. Measured injection characterization points (circles) and the corresponding curve fits (dashed lines).

expressed as

$$\log\left(\frac{\Delta I_{\mathrm{s}}}{I_{\mathrm{s0}}}\right) = g\left(\log\left(\frac{I_{\mathrm{s0}}}{I_{\mathrm{u}}}\right)\right) V_{\mathrm{ds}} + f\left(\log\left(\frac{I_{\mathrm{s0}}}{I_{\mathrm{u}}}\right)\right), \tag{2.15}$$

where $g(\cdot)$ and $f(\cdot)$ are weakly linear functions when the transistor is in the subthreshold region and are nonlinear when the transistor is in the above-threshold region. Before the programming process, a floating-gate characterization process is performed. In the characterization process, V_{ds} and I_{s0} are given and ΔI_{s} can be measured. Thus, $g(\log(I_{\mathrm{s0}}/I_{\mathrm{u}}))$ and $f(\log(I_{\mathrm{s0}}/I_{\mathrm{u}}))$ can be regressed by high-order polynomial functions. After the characterization process, the resultant polynomial functions, $\hat{f}(\log(I_{\mathrm{s0}}/I_{\mathrm{u}}))$ and $\hat{g}(\log(I_{\mathrm{s0}}/I_{\mathrm{u}}))$, are obtained. In the programming process, with these polynomial functions, the appropriate V_{ds} value for injection can be predicted by

$$V_{\mathrm{ds}} = \frac{\log\left(\frac{\Delta Is}{I_{\mathrm{s0}}}\right) - \hat{f}\left(\log\left(\frac{I_{\mathrm{s0}}}{I_{\mathrm{u}}}\right)\right)}{\hat{g}\left(\log\left(\frac{I_{\mathrm{s0}}}{I_{\mathrm{u}}}\right)\right)}, \tag{2.16}$$

where I_{s0} is the given starting point and I_{s} is the target value.

Measured and regression results for the injection characteristics are compared in Figure 2.8. Here, the polynomial functions are cubic and the pulse width is fixed at 200μsec.

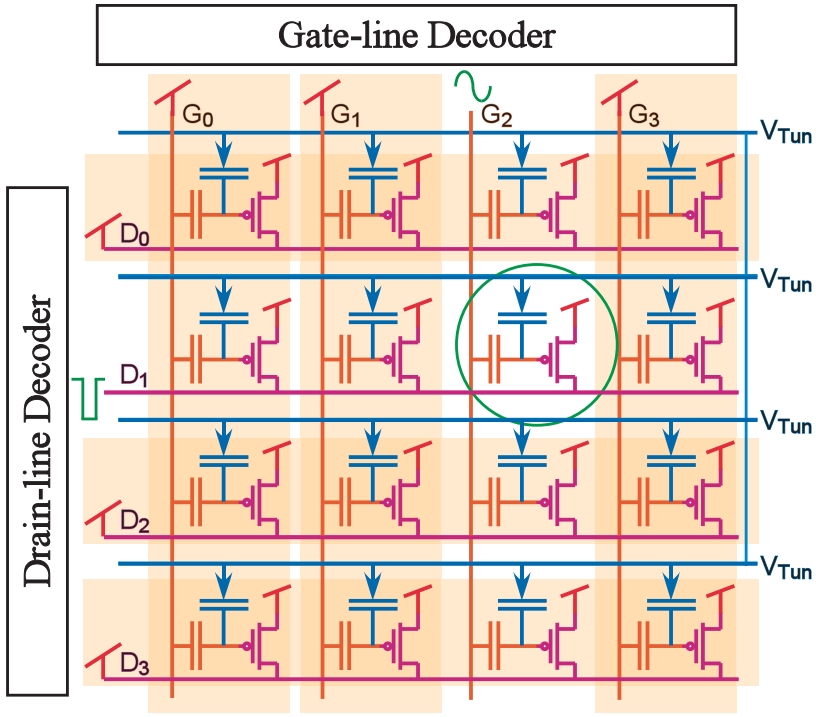


Figure 2.9. Floating-gate array programming.

10 different values of V_{ds} ranging from 5.6V to 6.5V and 30 channel current levels ranging from 20nA to 20μA are used to obtain the curve fits for each curve. To avoid overshooting the target value, we always apply slightly shorter and smaller pulses to the drain than the predicted values. Therefore, despite the mismatches and the discrepancy between the curve fits and the measured data, the current level of the floating-gate transistor approaches the target value asymptotically. The current level can be programmed within 0.5% of the target.

To program an array of floating-gate transistors, the devices are arranged into a two-dimensional array in programming mode, as shown in Figure 2.9. All the source terminals are connected to V_{DD}. The gate coupling voltages and the drain voltages are switched between a V_{DD} and the outputs of digital-to-analog converters. The decoders select which device is connected to the DAC voltages. Because two conditions, sufficient channel current and a high channel-to-drain field, are required for injection, the unselected columns (or rows) can be deactivated by applying V_{DD} to the corresponding gate-lines (or drain-lines)

so that there is no current through (or no field across) the devices for injection. In this manner, each floating-gate transistor can be isolated from others and can be programmed individually.

In this topology, the maximum number of electrons that can be injected into a floating-gate transistor is limited by the current isolation rule. If there are too many electrons on a floating-gate transistor, the source current is not negligible when the external coupling voltage is V_{DD}. If the drain-line is pulsed to program another floating-gate transistor in the same row, the unselected floating-gate transistor will also be injected. To avoid this problem, an extra pFET transistor can be used as a switch to shut the source current down completely when a row is not selected.

CHAPTER 3

A CHARGE-BASED LOW-POWER CAPACITIVE SENSING INTERFACE CIRCUIT

This chapter presents a low-power approach to capacitive sensing that can achieve a high signal-to-noise ratio. The circuit is composed of a capacitive feedback charge amplifier and a charge adaptation circuit. Without the adaptation circuit, the charge amplifier only consumes 1 μW and achieves an SNR of 69.34 dB in the audio band. An adaptation scheme using Fowler-Nordheim tunneling and channel hot electron injection mechanisms to stabilize the DC output voltage is demonstrated. This scheme provides a low-frequency corner at 0.2 Hz. The measured noise spectra show that this slow adaptation does not degrade the circuit performance. The DC path can also be provided by a large feedback resistance without causing extra power consumption. A charge amplifier with a MOS-bipolar nonlinear resistor feedback scheme is interfaced with a capacitive micromachined ultrasonic transducer to demonstrate the feasibility of this approach to ultrasound applications.

3.1 Challenges in Capacitive Sensing

Capacitive transduction is widely used in sensory microsystems to detect force, pressure, as well as position, velocity, or acceleration of a moving object. In a typical two-chip hybrid approach, as shown in Figure 3.1, there is a parasitic capacitance at the interconnect

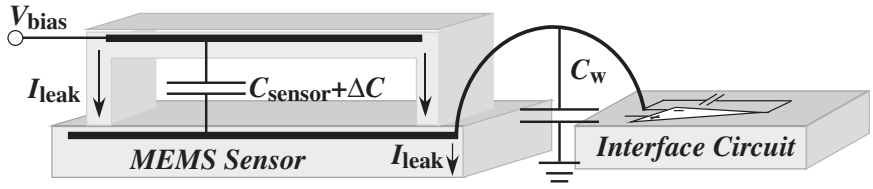


Figure 3.1. The block diagram of a two-chip hybrid approach.

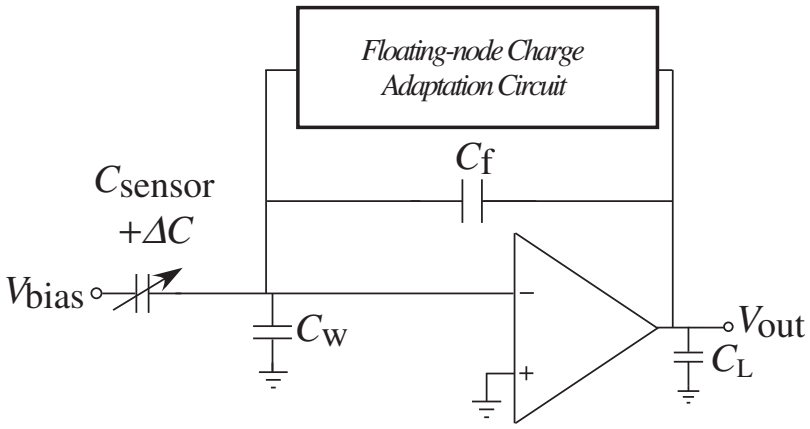


Figure 3.2. The block diagram of the capacitive sensing.

resulting from bonding wires and pads. The parasitic capacitance and the static sensor capacitance are usually much larger than the varying capacitance to be sensed. Additionally, some unpredictable and undesired leakage currents exist at the sensor-electronics interface. Designing a capacitive sensing interface circuit that has a large dynamic range and high sensitivity is not a trivial task. It becomes more challenging to design interface circuits for low-power applications.

In this chapter, a new approach to sensing capacitive changes is proposed. As shown in Figure 3.2, the circuit is composed of a capacitive feedback charge amplifier and a charge adaptation circuit. The charge amplifier operates continuously in time and can be viewed as a capacitive circuit. To cope with the charge and leakage currents at the floating node, methods based on floating-gate circuit techniques [6, 7, 10] are employed to adapt the charge. The analysis given in this chapter starts from a capacitive feedback charge amplifier without the charge adaptation circuit. The complete analysis with the charge adaptation will be presented in Chapter 4.

3.2 Capacitive Sensing Charge Amplifier

The small-signal model of a capacitive feedback charge amplifier without the charge-adaptation circuit is shown in Figure 3.3. The amplifier is modeled as a first-order system. The topology can be a simple cascode operational transconductance amplifier (OTA), as

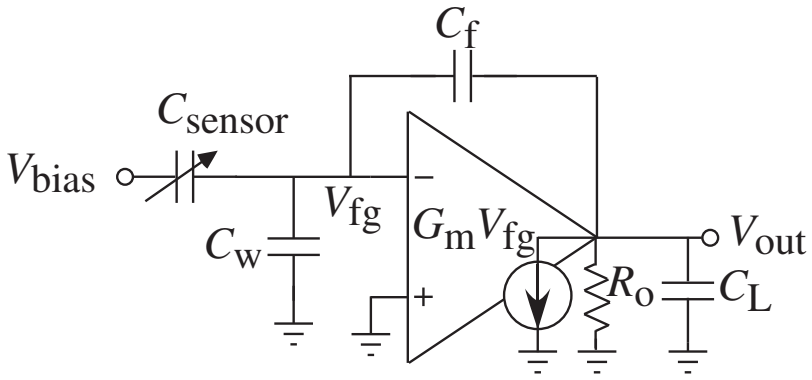


Figure 3.3. The small-signal model of a capacitive sensing charge amplifier.

shown in Figure 3.4, a folded cascode amplifier, or a cascode common-source amplifier. If the amplifier has two stages, although similar results can be derived, the dominant pole will depend on the compensation capacitance and the power consumption will be larger.

3.2.1 Transfer function

The DC output voltage can be expressed as

$$V_{out} = -\frac{V_{bias} C_{sensor} + Q}{C_f(1+\rho)}, \tag{3.1}$$

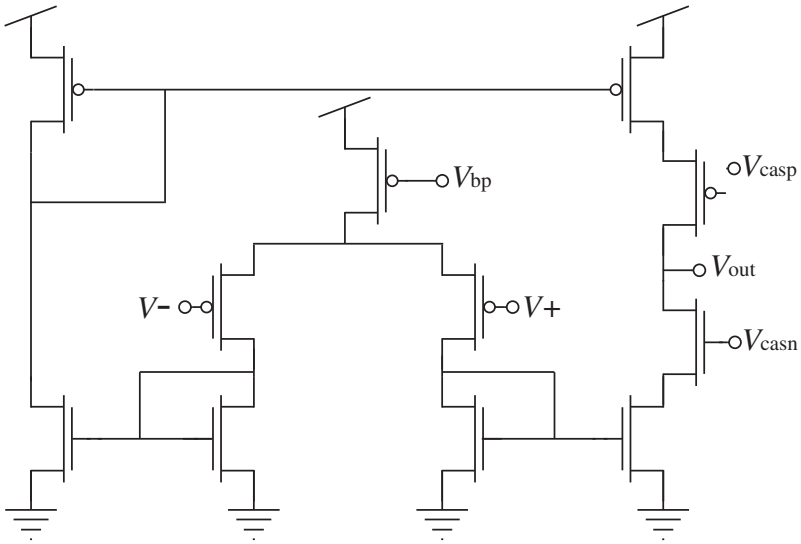


Figure 3.4. The schematic of a single-stage cascode operational transconductance amplifier.

where $\rho = C_T/(C_f A_v)$, $C_T = C_{sensor} + C_w + C_f$, $A_v = G_m R_o$, and Q is the charge on the floating node. Usually the amplifier gain should be designed high enough so that

$$A_v \gg C_T/C_f, \tag{3.2}$$

and

$$\rho \ll 1. \tag{3.3}$$

In this case, (3.1) can be approximated as

$$V_{out} \approx -\frac{V_{bias}C_{sensor} + Q}{C_f} \text{ as } \rho \to 0. \tag{3.4}$$

The DC level of V_{out} can be set at halfway between the rails either by adjusting the non-inverting terminal voltage if the floating-gate charge is fixed, or by adjusting the floating-gate charge according to the output voltage.

The variation of the output voltage is proportional to the variation of the sensor capacitance and can be expressed as

$$\Delta V_{out} = -\frac{V_{bias}}{C_f(1+\rho)} \cdot \Delta C_{sensor}. \tag{3.5}$$

The circuit can achieve high sensitivity if a large value of V_{bias} and a small value of C_f are used. A recorded music waveform from a version of the capacitive sensing charge amplifier that is interfaced with an audio MEMS microphone is shown in Figure 3.5.

If the floating-node voltage is regulated by the feedback and is assumed to be constant, the transfer function of the capacitive sensing charge amplifier can be expressed as

$$\frac{V_{out}(s)}{C_{sensor}(s)} = -\frac{V_{bias}}{C_f(1+\rho)} \cdot \frac{1 - sC_f/G_m}{1 + s\tau}, \tag{3.6}$$

where τ is the time constant of the circuit and is given as

$$\tau = \frac{1}{\omega_{3dB}} = \frac{C_{eff}}{G_m(1+\rho)}, \tag{3.7}$$

where $C_{eff} = \left(C_o C_T - C_f^2\right)/C_f$, and $C_o = C_L + C_f$. Because both C_{sensor} and C_L are usually larger than C_f, the zero caused by the capacitive feedthrough is at a much higher frequency

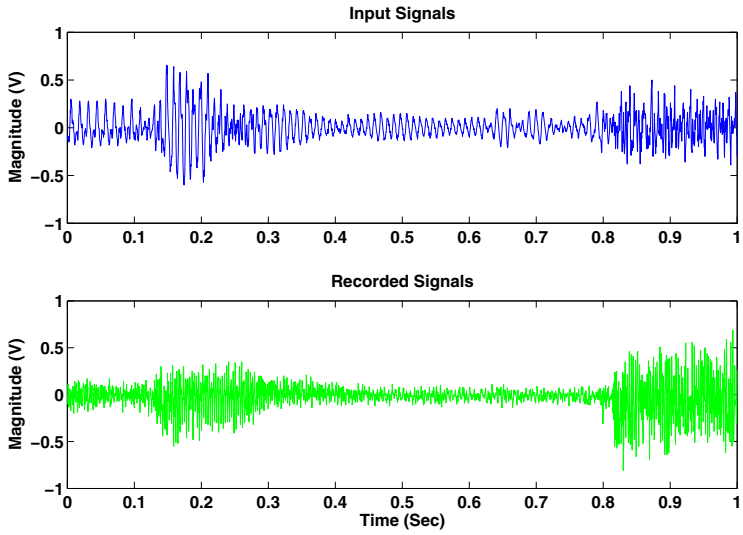


Figure 3.5. Measured waveform from a capacitive sensing charge amplifier.

than the amplifier bandwidth. If the amplifier gain is large enough ($A_\mathrm{v} \gg C_\mathrm{T}/C_\mathrm{f}$), the transfer function can be approximated as

$$\frac{V_\mathrm{out}(s)}{C_\mathrm{sensor}(s)} = -\frac{V_\mathrm{bias}}{C_\mathrm{f}} \cdot \frac{1 - sC_\mathrm{f}/G_\mathrm{m}}{1 + sC_\mathrm{eff}/G_\mathrm{m}}. \tag{3.8}$$

3.2.2 Noise analysis

The output-referred noise power can be calculated from the small-signal model shown in Figure 3.6a. If V_fg and V_out are related by a capacitive divider, the small signal model can be further simplified, as shown in Figure 3.6b, where $R_\mathrm{x} = C_\mathrm{T}/(C_\mathrm{f}G_\mathrm{m})$. The output-referred voltage noise can be expressed as

$$\hat{V}^2_\mathrm{out,total} = \frac{\tilde{i}^2_\mathrm{o}}{4G_\mathrm{m}} \cdot \frac{C_\mathrm{T}}{C_\mathrm{f}C_\mathrm{o}} \cdot \frac{1}{1+\rho} \tag{3.9}$$

$$= \frac{nqU_\mathrm{T}}{2\kappa} \cdot \frac{C_\mathrm{T}}{C_\mathrm{f}C_\mathrm{o}} \cdot \frac{1}{1+\rho}. \tag{3.10}$$

In (3.10), the differential pair transistors are assumed operating in the subthreshold region and

$$\tilde{i}^2_\mathrm{o} = \frac{2}{\kappa} nqU_\mathrm{T}G_\mathrm{m} \quad (A^2/\mathrm{Hz}), \tag{3.11}$$

where κ is the subthreshold slope factor of a MOS transistor, n is the effective number of noisy transistors, q is the charge of an electron, U_T is the thermal voltage, and G_m is the transconductance of the transistor in the differential pair. This result is consistent with that in [11], where the transistors are assumed to operate in the above-threshold region.

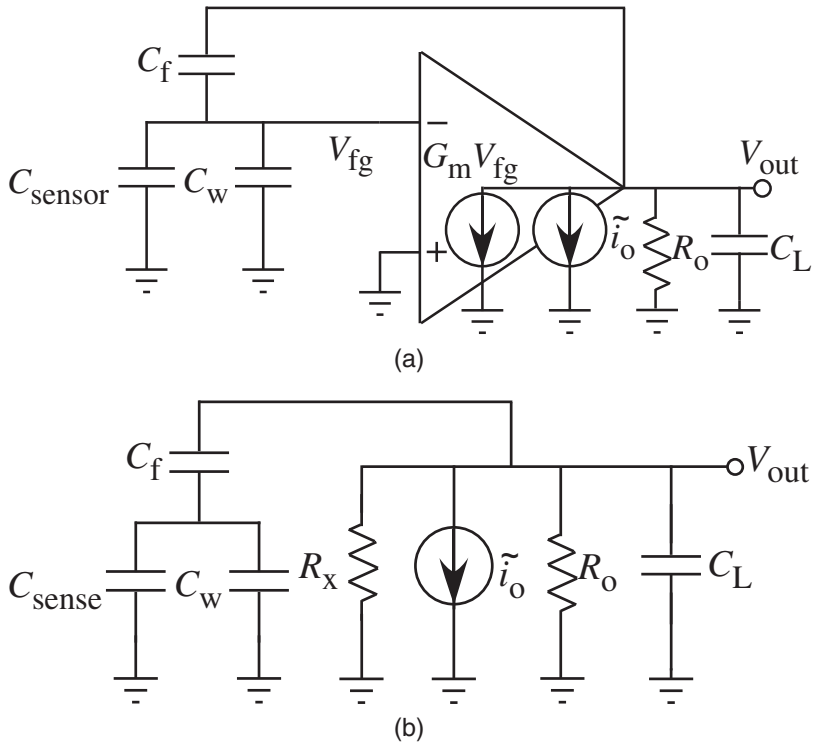


Figure 3.6. The small-signal models of the capacitive sensing charge amplifier for noise analysis.
(a) The small-signal model for noise analysis
(b) Simplified small-signal model for noise analysis

3.2.3 Maximum dynamic range

The nonlinearity of the circuit is assumed to come from the voltage-controlled current source of the transconductance amplifier. When the input voltage is smaller than or equal to the maximum input linear voltage, $\Delta V_{lin,max}$, the output current is linear in the input voltage with some tolerable distortion. For example, if the OTA shown in Figure 3.4 is employed in the charge amplifier and the differential pair operates in the subthreshold region, the I-V

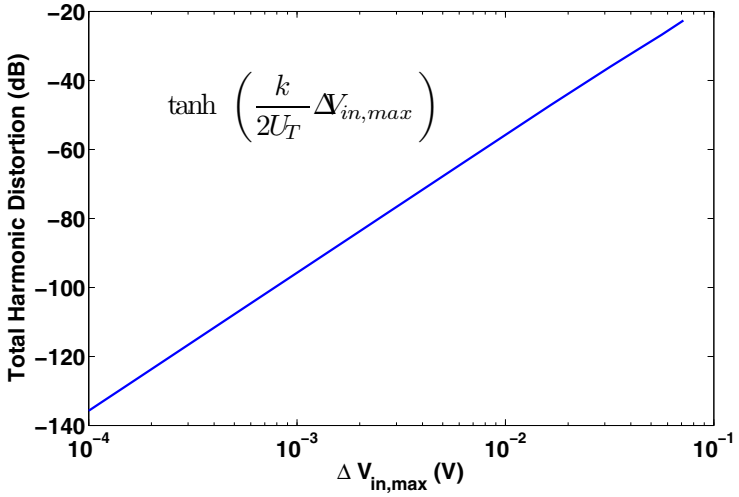


Figure 3.7. Nonlinearity of an OTA.

relation can be expressed as

$$I_{\text{out}} = I_{\text{b}} \tanh\left(\frac{\kappa V_{\text{d}}}{2U_{\text{T}}}\right) \tag{3.12}$$

$$\approx I_{\text{b}} \frac{\kappa \Delta V_{\text{fg}}}{2U_{\text{T}}}, \tag{3.13}$$

where I_{b} is the tail current and V_{d} is the OTA input differential voltage. The hyperbolic-tangent function gives rise to the nonlinearity of the circuit. The relation between the total harmonic distortion and the maximum input linear range of the differential pair is plotted in Figure 3.7. If $V_{\text{d}} \leq \Delta V_{\text{lin,max}} = 8\,\text{mV}$, the approximation of

$$G_{\text{m}} \approx \frac{\kappa I_{\text{b}}}{2U_{\text{T}}} \tag{3.14}$$

brings about $-60\,\text{dB}$ total harmonic distortion when $U_{\text{T}} = 25\,\text{mV}$ and $\kappa = 0.7$ are used in (3.12).

The maximum output linear range, $\Delta V_{\text{out,max}}$, can be obtained from the transfer function of V_{fg} to V_{out} as

$$\begin{aligned} \left|\Delta V_{\text{out,max}}(j\omega)\right| &= \left|\frac{(j\omega)C_{\text{f}} - G_{\text{m}}}{(j\omega)C_{\text{o}} + G_{\text{m}}/A_{\text{v}}} \Delta V_{\text{fg}}(j\omega)\right| \\ &\leq \left|\frac{(j\omega)C_{\text{f}} - G_{\text{m}}}{(j\omega)C_{\text{o}} + G_{\text{m}}/A_{\text{v}}}\right| \Delta V_{\text{lin,max}}. \end{aligned} \tag{3.15}$$

A normalized variable, η, can be defined as

$$\eta = \omega/\omega_{3dB} = \omega\tau. \tag{3.16}$$

The expression of (3.15) can be rewritten as

$$\left|\Delta V_{\text{out,max}}(\eta)\right| \leq \frac{C_{\text{T}}}{C_{\text{f}}} \cdot \frac{\Delta V_{\text{lin,max}}}{\sqrt{\eta^2(1+\rho)^2+\rho^2}}. \tag{3.17}$$

If the gain of the amplifier and the bandwidth of the circuit are both infinite (i.e. $\rho = 0$ and $\eta = 0$), from (3.17), then the maximum output linear range is also infinite. The circuit is completely linear in this ideal scenario, if the limitations from supply rails do not come into play.

In reality, the amplifier has a high but finite gain ($0 < \rho \ll 1$) and, to save power, the operating frequency is usually close to and within the bandwidth ($\eta \approx 1$). In this case, we can approximate (3.17) as

$$\left|\Delta V_{\text{out,max}}(\omega)\right| \leq \frac{G_{\text{m}}}{C_{\text{o}}} \cdot \frac{\Delta V_{\text{lin,max}}}{\omega}. \tag{3.18}$$

From (3.18), the output linear range can be increased by increasing the G_{m} or by decreasing the load capacitance. If the $\Delta V_{\text{out,max}}$ is defined as the maximum output linear range in the worst scenario within the bandwidth, we can have

$$\begin{aligned}\Delta V_{\text{out,max}} &\equiv \inf_{0<\omega\leq\omega_{3dB}} \frac{G_{\text{m}}}{C_{\text{o}}} \cdot \frac{\Delta V_{\text{lin,max}}}{\omega} \\ &= \frac{G_{\text{m}}}{C_{\text{o}}} \cdot \frac{\Delta V_{\text{lin,max}}}{\omega_{3dB}} \\ &= \frac{C_{\text{T}}}{C_{\text{f}}} \cdot \Delta V_{\text{lin,max}},\end{aligned} \tag{3.19}$$

which is the lower bound of the maximum output linear range.

3.2.4 Signal-to-noise ratio

The lower bound of the circuit SNR can be obtained from (3.19) and (3.10) and can be expressed as

$$SNR \geq C_{\text{eff}} \cdot \frac{2\kappa {\Delta V_{\text{lin,max}}}^2}{nqU_{\text{T}}}. \tag{3.20}$$

The SNR can be expressed as a function of the ratio of the operating frequency to the 3dB frequency, η, and can be written as

$$SNR(\eta) = \frac{C_T C_o}{C_f} \cdot \frac{2\kappa\Delta V_{lin,max}{}^2}{nqU_T} \cdot \frac{1+\rho}{\eta^2(1+\rho)^2+\rho^2} \tag{3.21}$$

$$\approx C_{eff} \cdot \frac{2\kappa\Delta V_{lin,max}{}^2}{nqU_T} \cdot \frac{1+\rho}{\eta^2(1+\rho)^2+\rho^2} \tag{3.22}$$

$$\approx \frac{G_m}{\omega_{3dB}} \cdot \frac{2\kappa\Delta V_{lin,max}{}^2}{nqU_T} \cdot \frac{1+\rho}{\eta^2(1+\rho)^2+\rho^2}. \tag{3.23}$$

From (3.21), if the ratio of the operating frequency to the 3dB frequency is constant, the SNR can be improved by increasing C_T, the total capacitance seen from the floating node, by increasing the output capacitance, C_o, or by decreasing C_f, the feedback capacitance. To maintain the same bandwidth, more power needs to be consumed to increase the value of G_m.

The SNR can also be expressed as a function of the frequency as

$$SNR(\omega) \approx \frac{G_m^2}{C_{eff}} \cdot \frac{2\kappa\Delta V_{lin,max}{}^2}{nqU_T} \cdot \frac{1}{\omega^2(1+\rho)} \tag{3.24}$$

From (3.24), for a given operating frequency, the SNR can be increased by increasing G_m or by decreasing C_{eff}. There exists a contradiction between (3.22) and (3.24) in the effect of C_{eff} on the SNR. In (3.24), the improvement of the SNR is caused by the increase of the circuit bandwidth. If the circuit bandwidth is fixed, (3.24) can be expressed as

$$SNR(\omega) \approx C_{eff} \cdot \omega_{3dB}{}^2 \cdot \frac{2\kappa\Delta V_{lin,max}{}^2}{nqU_T} \cdot \frac{1}{\omega^2(1+\rho)}, \tag{3.25}$$

where C_{eff} has the same effect as it does in (3.22).

3.2.5 Measurement results

A version of the capacitive sensing circuit was fabricated in a 0.5 μm double-poly CMOS process. A MEMS microphone sensor fabricated in a Sandia National Laboratory's silicon based SwIFT-Lite process [12] is used to interface with the circuit. The typical value of the sensor capacitance is in the range of pico farads. Measurement setup is shown in Figure 3.8

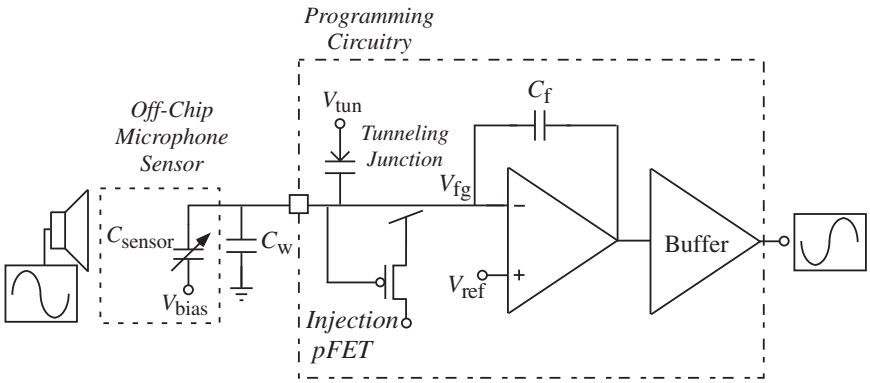


Figure 3.8. The setup for audio measurements.

and micrographs of the chip and sensors are shown in Figure 3.9. An ultra-thin card type speaker is used as an acoustic signal source. A tunneling junction and an indirect injection transistor are integrated on the chip as parts of the floating-node charge adaptation circuit. To measure the characteristics without any charge adaptation circuit, both tunneling and drain voltages are tied to V_{DD}. The floating-node voltage can settle slowly to an equilibrium value, which is very sensitive to the environmental electromagnetic interference. To avoid the perturbation of the floating-node voltage, the chip and the sensor are placed inside a shielding metal box. In this setup, the non-inverting terminal voltage can be carefully adjusted to set the output DC level to halfway between the rails.

The spectrum of a output waveform with 1V magnitude at 1kHz with -38 dB total harmonic distortion is shown in Figure 3.10. The distortion comes from the offset, from the cascoded output stage of the amplifier, as well as from the nonlinearity of the speaker and of the MEMS sensor. The noise spectrum of the sensing circuit without the MEMS sensor is also shown in the same plot. The output noise depends on the input capacitance as analyzed in (3.10). To have a better idea of the circuit noise performance when it is interfaced with a MEMS sensor, the sensor is replaced with a 2 pF linear capacitor. By varying the tail current of the OTA, three noise spectrums that have power consumption of 1 μW, 0.23 μW, and 0.13 μW are measured and plotted in Figure 3.11. As expected from (3.7) and (3.10), increasing the value of G_m results in a higher bandwidth with the cost of

more power consumption, but the total output thermal noise is independent of G_m. If the sensing circuit is followed by a low-pass filter that limits the noise bandwidth inside the audio band (i.e. 20Hz to 20kHz with uniform weighting), the resulting output noise can be reduced by burning more power.

The integrated total output noise over the entire bandwidth is 570 μV_{rms}. The integrated thermal noise is 520 μV_{rms}, which is slightly higher than 370 μV_{rms}, the value estimated from (3.10). When the power consumption is 1 μW, the flicker noise corner is around 2kHz, and the integrated flicker noise in that range is 225 μV_{rms}. The integrated total output noise in the audio band is 341 μV_{rms}. The corresponding minimum detectable capacitance variation in the audio band is 83 aF that is estimated from (3.8) using the parameters listed in Table 3.1. The capacitance sensitivity in the audio band is 0.59 aF/$\sqrt{\text{Hz}}$ and the minimum detectable displacement is 20.76×10^{-4} Å/$\sqrt{\text{Hz}}$. The SNR of the circuit is 64.88 dB over the entire bandwidth and is 69.34 dB in the audio band with the power consumption of 1 μW.

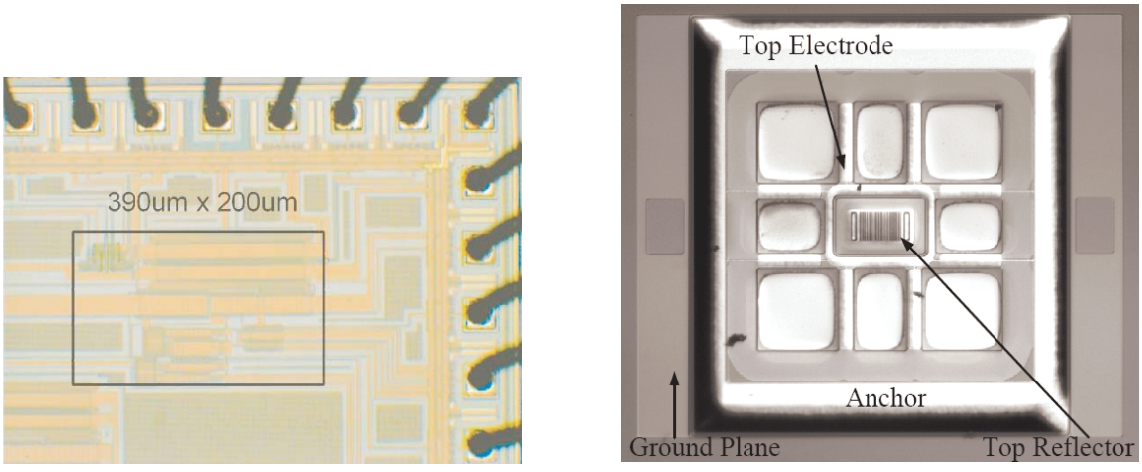


Figure 3.9. The micrographs of the charge amplifier and the MEMS microphone sensor used in the measurement.
(a) A die micrograph of a version of the capacitive feedback charge amplifier
(b) The micrographs of the MEMS microphone sensor

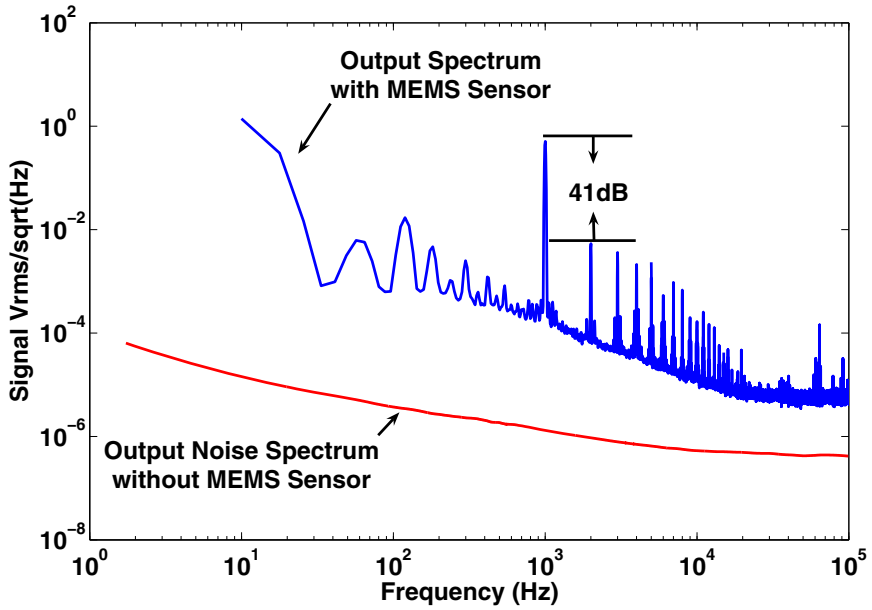


Figure 3.10. The measured output signal and noise spectrums.

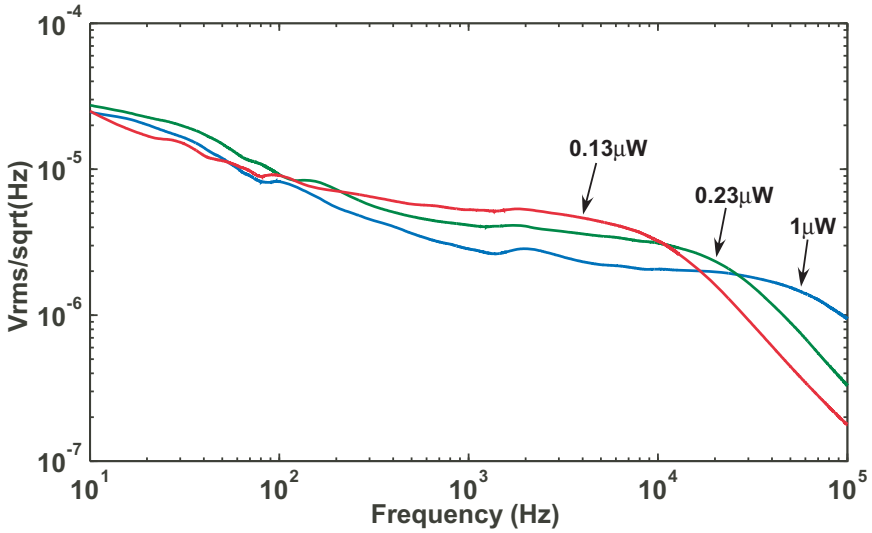


Figure 3.11. Noise spectrums of a charge amplifier with a linear input capacitor.

3.3 Floating-Gate-Based Charge Adaptation

In the previous section, it has been shown that using a capacitive feedback charge amplifier to sense capacitance change can achieve a high SNR with low power consumption. The performance will be compared with other approaches in the subsequent section. Although the output DC level can be set by adjusting the voltage at the non-inverting terminal, it is

not stable. Without the shielding metal box, the output voltage is prone to being saturated to the supply rails because the floating-node voltage is very sensitive to the electromagnetic interference in the testing environment. In this section, floating-gate techniques are employed to stabilize the output DC level.

3.3.1 Audio applications

The schematic used to demonstrate the capacitive sensing circuit with a charge adaptation circuit is shown in Figure 3.12. To reduce the leakage current caused by the ESD protection circuitry, the sensor is bonded to the chip with electronics via a bare pad. The leakage current at the connection can be integrated directly by the charge amplifier. Measured leakage current is between 5 fA and 1 pA depending on the biasing voltage and the supply rails.

A tunneling junction and an injection transistor are integrated on chip to compensate for

Table 3.1. Parameters and Measurement Results From A Capacitive Sensing Charge Amplifier

CIRCUIT PARAMETERS	
Area	$390 \times 200\,\mu m^2$
Power Supply	3.3V
Amplifier Power Consumption	$1\,\mu W$
Open-Loop Gain	80 dB
Sensor Bias Voltage V_{bias}	3.3V
Feedback Capacitance C_f	800 fF
MEASUREMENT RESULTS AND PERFORMANCE	
Measured Leakage Current	5 fA to 1 pA
Output-referred Noise (Audio Band)	$341\,\mu V_{rms}$
Signal to Noise Ratio (Audio Band)	69.34 dB
Min. Detectable Capacitance (Audio Band)	83 aF
Capacitance Sensitivity @1kHz	$0.59\,aF/\sqrt{Hz}$
Min. Detectable Displacement @1kHz	$20.76 \times 10^{-4}\,Å/\sqrt{Hz}$

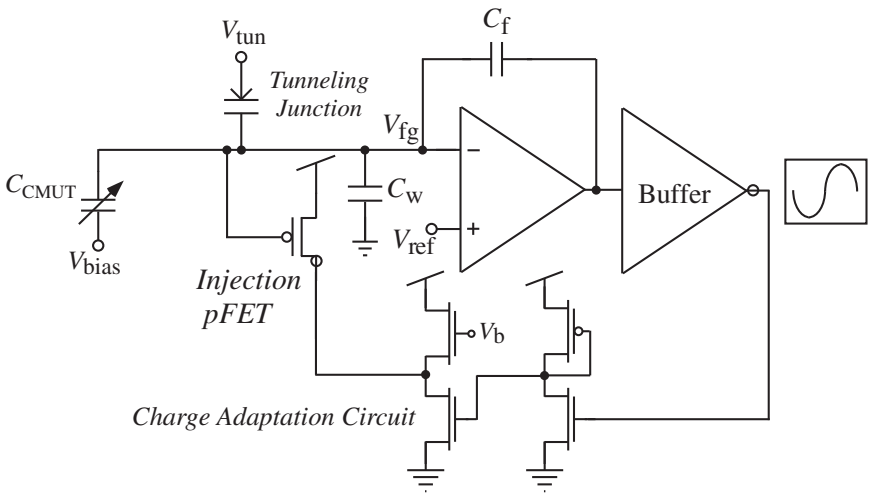


Figure 3.12. Setup for the measurement with charge adaptation scheme.

the leakage current. The injection current can be controlled by the transistor's drain voltage. One of many possible ways to control the drain voltage according to the output DC level is shown in Figure 3.12. In this adaptation scheme, the tunneling voltage is kept constant and only the injection current varies to compensate for the leakage current. The adaptation scheme can auto-zero the output DC voltage to halfway between the rails without affecting the performance of the circuit. The dynamics of these two mechanisms are detailed in [13].

To have an enough field to generate the injection current, the supply rail is raised to 6.5 V and the externally applied tunneling voltage is 13 V. As shown in Figure 3.13, the output voltage is adapted to halfway between the rails after an upward or a downward step is applied to the sensor bias voltage. The extracted time constants are 5 seconds and 30 seconds, respectively. This implies that the effective resistance caused by the adaptation scheme is on the scale of 10^{12} Ω. Increasing the tunneling voltage results in a faster adaptation rate because of larger adaptation current.

The noise spectra with and without this adaptation scheme are compared in Figure 3.14. In both cases, the circuit is interfaced with the MEMS microphone sensor. It is shown that this adaptation scheme does not degrade the noise performance over the frequency band of interest. By introducing the tunneling-injection adaptation current, the low frequency

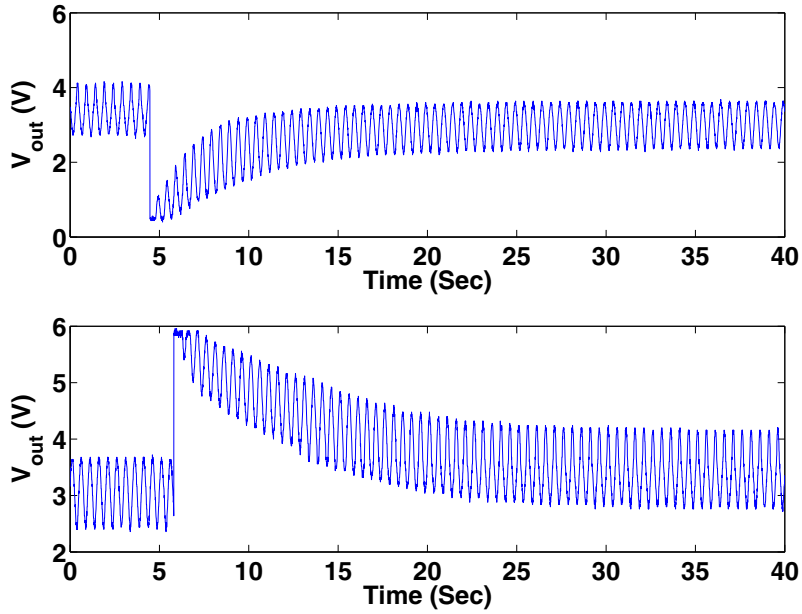


Figure 3.13. The step responses of the capacitive sensing charge amplifier with a tunneling-injection adaptation scheme.

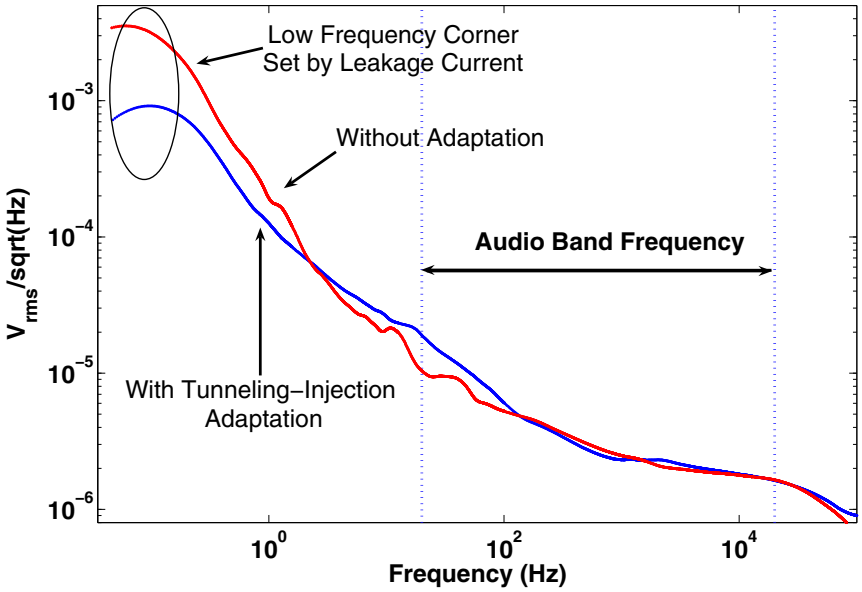


Figure 3.14. The measured noise spectra with and without the adaptation circuit.

corner moves slightly toward higher frequencies. The corner frequency is at 0.2 Hz and is consistent with the extracted time constants from Figure 3.13.

The power consumption in the charge adaptation circuit in Figure 3.12 is in the range

of milli-watts because the adaptation circuit is made up of discrete components. If these transistors are also integrated on chip and are designed to be long, the power consumption can be much lower. If the charge adaptation is implemented by using the topology of an autozeroing floating-gate amplifier, as proposed in [7], no extra power is consumed for the charge adaptation.

The externally applied tunneling voltage can be generated on chip using a charge pump circuit. In this case, the tunneling voltage, as well as the tunneling current, can be adjusted by the input voltage or the clock frequency of the charge pump circuit. As a result, the adaptation time constant can also be tuned on chip. Since the tunneling junction is small and the tunneling current is in the range of pico-amps, the extra cost of the silicon area and the power consumption for the charge pump circuit, including the clock generator, diodes, and capacitors, is usually low. In some applications, if the medium of the sensor is leaky and the leakage current is large enough to provide a fast adaptation time constant, the tunneling current is not necessary for charge adaptation. In this case, the circuit to generate the tunneling voltage, as well as the tunneling junction, can be avoided altogether.

3.3.2 Ultrasonic applications

The charge amplifier approach has been applied to sensing capacitive micromachined ultrasonic transducers (CMUTs) that have been recently developed for ultrasound imaging [14, 15]. The measured leakage current of a CMUT device can be up to 500 pA. Since the operating frequency is high, besides the tunneling-injection adaptation scheme, a series of MOS-bipolar nonlinear resistors can be used to provide the DC path to the floating node.

A MOS-bipolar nonlinear resistor is a *p*MOS transistor with the connections from the gate to the drain and from the bulk to the source. It exhibits a very large value of resistance (exceeding 10^{12} Ω) when the cross voltage is close to zero. The MOS-bipolar nonlinear resistor has been used in quasi-floating-gate transistor circuits [10] and the neural recording application [16]. To extend the output linearity, two nonlinear resistors in series are used to provide the DC path to the floating node.

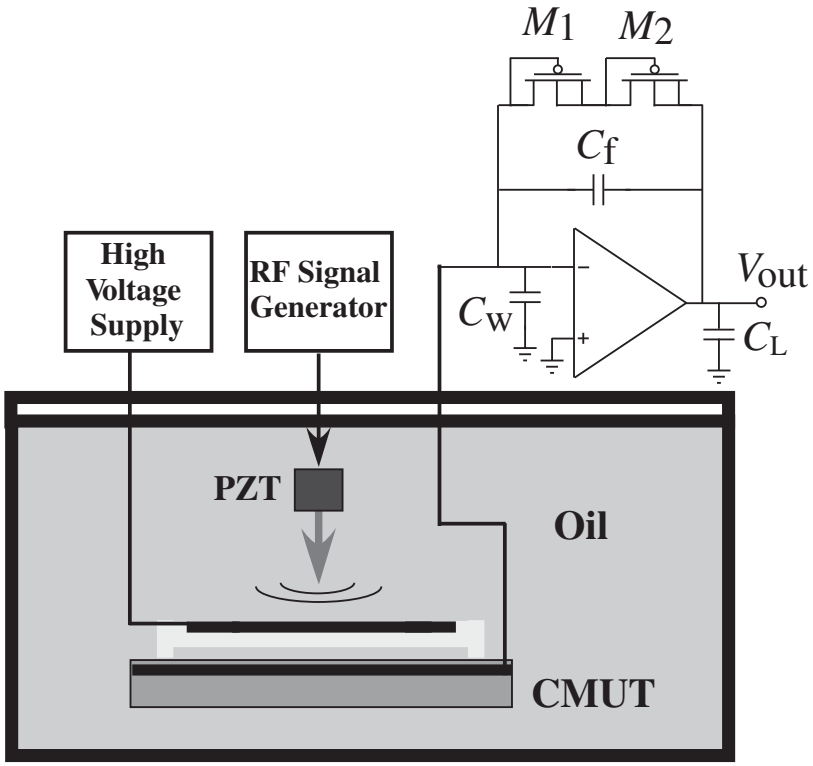


Figure 3.15. Setup for the ultrasonic measurement.

The setup for ultrasonic measurements is illustrated in Figure 3.15. A piezo transducer is used to generate plane waves at 1MHz using 16V peak 5 cycle tone bursts at its input. A CMUT receiver is biased to 90V DC at one of its terminals. The other terminal is connected to a sensing amplifier. The CMUT and the piezo device are submerged in the oil during the measurement. The capacitance of the CMUT sensor is about 2 pF and the maximum variation is about 1%. The resultant pulse-echo response is shown in Figure 3.16. The initial, highly distorted signal is due to the electromagnetic feedthrough. After 15 microseconds, the first acoustic signal arrives from the piezo transducer to the CMUT receiver. A corresponding distance is 2.2 cm in the oil, as expected. By changing this distance and the relative alignment of the piezo device and the CMUT receiver, the received signal and multiple echoes change drastically, again as expected from an ultrasound transmission experiment. Some important parameters for the CMUT measurement are listed in Table 3.2.

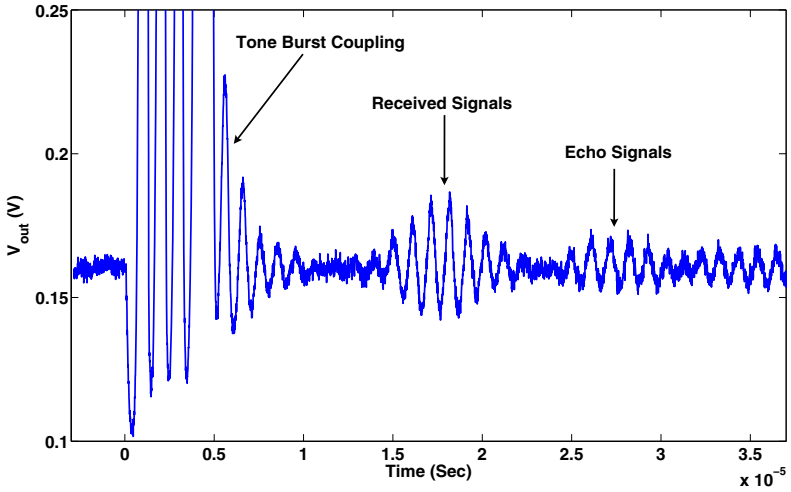


Figure 3.16. Measured pulse-echo waveform from an ultrasonic transducer.

Table 3.2. CMUT Measurement Parameters

Amplifier Power Supply	3.3V
CMUT Bias Voltage	$90V$
CMUT Leakage Current	500 pA
CMUT Capacitance	2 pF
Piezo Transducer Frequency	$1MHz$
CMUT and Piezo Transducer Spacing	$2.2cm$

3.4 Design Procedure

Given the specifications of the minimum detectable capacitance (ΔC_{min}), the bandwidth (ω_{3dB}), and the SNR, we try to optimize the current consumption (I_{b}), the feedback capacitance (C_{f}), the load capacitance (C_{L}), and the total capacitance seen from the floating node ($C_{\text{T}} \approx C_{\text{sensor}} + C_{\text{w}}$). The known variables include the bias voltage for the sensing capacitor (V_{bias}) and the maximum input linear voltage of the transconductance amplifier ($\Delta V_{\text{lin,max}}$). It is also assumed that the maximum output linear range is not limited by the supply rails but only affected by the nonlinearity of the OTA.

The design starts from the sensitivity expression

$$\begin{aligned}\Delta C_{\text{min}}^2 &= \hat{V}_{\text{out,total}}^2 \cdot \left(\frac{C_\text{f}}{V_\text{bias}}\right)^2 \\ &= \frac{nqU_\text{T}}{2\kappa V_\text{bias}^2} \cdot \frac{C_\text{T}C_\text{f}}{C_\text{o}},\end{aligned} \tag{3.26}$$

where ΔC_{min} is the minimum detectable capacitance. From (3.26) and (3.25), the conditions for C_T and C_f/C_L can be given as

$$\frac{C_\text{f}C_\text{T}}{C_\text{L}} \leq \Delta C_{\text{min}}^2 \cdot \frac{2\kappa V_\text{bias}^2}{nqU_\text{T}}, \tag{3.27}$$

and

$$\frac{C_\text{T}C_\text{L}}{C_\text{f}} \geq SNR \cdot \frac{nqU_\text{T}}{2\kappa {\Delta V_{\text{lin,max}}}^2}. \tag{3.28}$$

Therefore,

$$\frac{C_\text{f}}{C_\text{L}} \leq \Delta C_{\text{min}} \cdot \frac{2\kappa V_\text{bias}\Delta V_{\text{lin,max}}}{nqU_\text{T}\sqrt{SNR}}, \tag{3.29}$$

and

$$C_\text{T} \geq \Delta C_{\text{min}} \cdot \sqrt{SNR} \cdot \frac{V_\text{bias}}{\Delta V_{\text{lin,max}}}. \tag{3.30}$$

Because only the ratio of C_f to C_L matters, reasonable and practical values can be chosen for these two capacitances.

The next step is to determine the current consumption for a given bandwidth. Assuming that the transistors in the OTA differential pair operate in the subthreshold region, (3.14) can be substituted into (3.7) and the required current consumption can be expressed as

$$I_\text{b} \geq \omega_{3dB} \cdot \frac{C_\text{T}C_\text{o}}{C_\text{f}} \cdot \frac{2U_\text{T}}{\kappa}. \tag{3.31}$$

For example, assuming SNR= 60 dB, $V_\text{bias} = 5$ V, and $\Delta V_{\text{lin,max}} = 10$ mV, if the desired minimum detectable capacitance is 10 aF with bandwidth of $40K$ Hz, the estimated C_f/C_L is 0.0875. If the load capacitance is 500 fF, the feedback capacitance should be less than 44 fF. The total capacitance from the floating node should be 5 pF. The estimated current consumption is about 0.4 μA.

3.5 Comparison

The switched-capacitor (SC) circuit shown in Figure 3.17a is commonly used to detect capacitive changes. Switches are inserted to reset the charge at the connecting node. Correlated double sampling (CDS) [17] techniques are also used [18, 19] to reduce the low-frequency noise and the DC offset. Issues like noise-folding, clock feed-through, and charge sharing need to be taken care of. For applications that require very high sensitivity, lock-in capacitive sensing is the most popular technique [20–22]. As shown in Figure 3.17b, the signals are modulated to high frequency and so the circuits consume lots of power and are very complicated. In either the SC or the lock-in approach, the sensing circuits process the entire charge on the sensing capacitor, instead of only the minute portion of it caused by the capacitance change. To cancel the effect of the large static capacitance, it is necessary to have differential capacitor structures to achieve a large dynamic range. However, the differential capacitor structure is not available in the capacitive MEM microphone sensor. Traditional approaches usually convert the capacitive current into a voltage that is then amplified in the subsequent stages. A self-biased JFET buffer shown in Figure 3.18a is the most commonly used interface circuit for electret condenser microphones (ECMs). However, JFETs are not available in the CMOS process and the gain is highly sensitive to the parasitic interconnect capacitance. In [23], the current through the JFET is sensed and amplified to improve the power supply rejection ration (PSRR) as shown in Figure 3.18b. Other approaches use diodes [24, 25] or a unity-gain feedback OTA [26] as a large resistor to convert the current to a voltage. The resulting voltage can be amplified directly or can be buffered and amplified in the next stage, as shown in Figure 3.18c–f. These approaches consume much less power compared with the SC and the lock-in techniques but the linearity is usually poor. The capacitive feedback charge amplifier approach to capacitive sensing consumes ultra-low power and achieves a large output dynamic range with high linearity. This work is compared with others in Table 3.3.

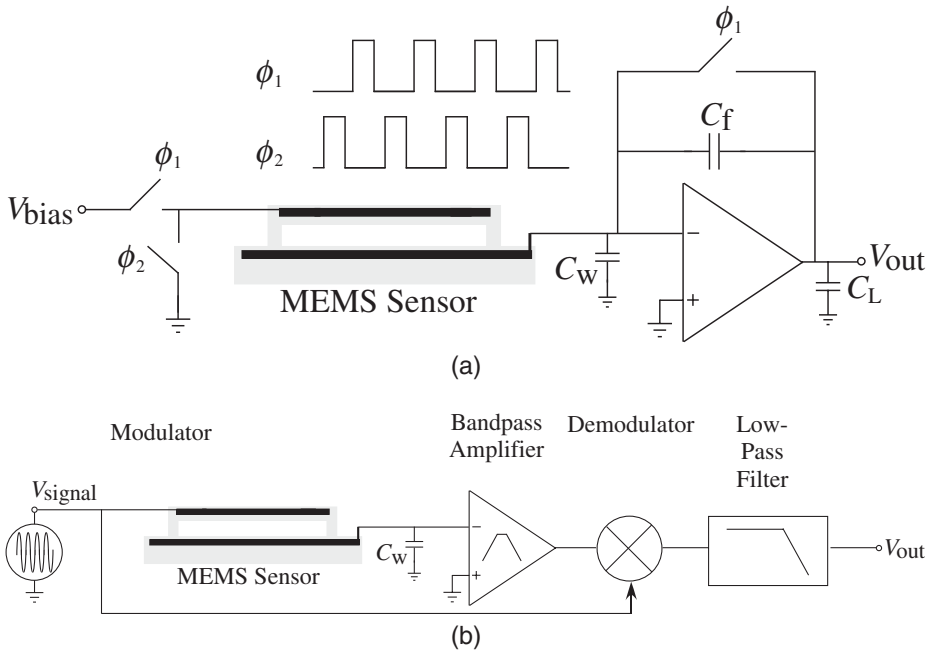


Figure 3.17. The block diagrams of the switched-capacitor and the lock-in approaches to capacitive sensing.
(a) The switch-capacitor approach
(b) The lock-in approach

3.6 Conclusion

The capacitive feedback charge amplifier is a simple topology. In this approach, the circuit amplifies the charge resulting from the capacitance change in the first stage without any I-V conversion or modulation. Therefore, this approach consumes very little power and achieves high linearity. The noise, the maximum dynamic range, and the SNR analysis are detailed in this chapter. An auto-zeroing method adopted from floating-gate circuit design techniques has been demonstrated to adapt the charge and to compensate for the leakage current at the floating node without affecting the performance of the circuit. Nonlinear resistors can also be used to provide a DC path to the floating node. These techniques have been demonstrated in the applications of sensing audio MEMS microphones and ultrasonic transducers.

Table 3.3. Performance Comparison

	[20]	[22]	[24]	[23]	[25]	[26]	This work
Method	Figure 3.17(b)	Figure 3.17(b)	Figure 3.18(c)	Figure 3.18(b)	Figure 3.18(f)	Figure 3.18(e)	Figure 3.9
Differential Cap.	Yes	Yes	No	No	No	No	No
Bandwidth (Hz)	2K	1 ~ 100	100 ~ 15K	100 ~ 10K	100 ~ 10K	100 ~ 10K	20 ~ 20K
Noise floor	1.12 aF	3.75 aF	27 dB SPL	4.8 μV_{rms}	25 dB SPL	30 μV_{rms}	83 aF
Output ref. noise	316 μV_{rms}	4.1 μV_{rms}	4.5 μV_{rms}	~ 100 μV_{rms}	5.6 μV_{rms}	300 μV_{rms}	341 μV_{rms}
$V_{out,max}$	0.13V	-	0.2V	0.51V	0.5V	0.2V	1V
THD	−60 dB	-	−20 dB	−40 dB	-	−50 dB	−38 dB
SNR (dB)	77	-	93	80	95	54	69
Power	30 mW	20 mW	150 μW	96 μW	60 μW	24 μW	1 μW

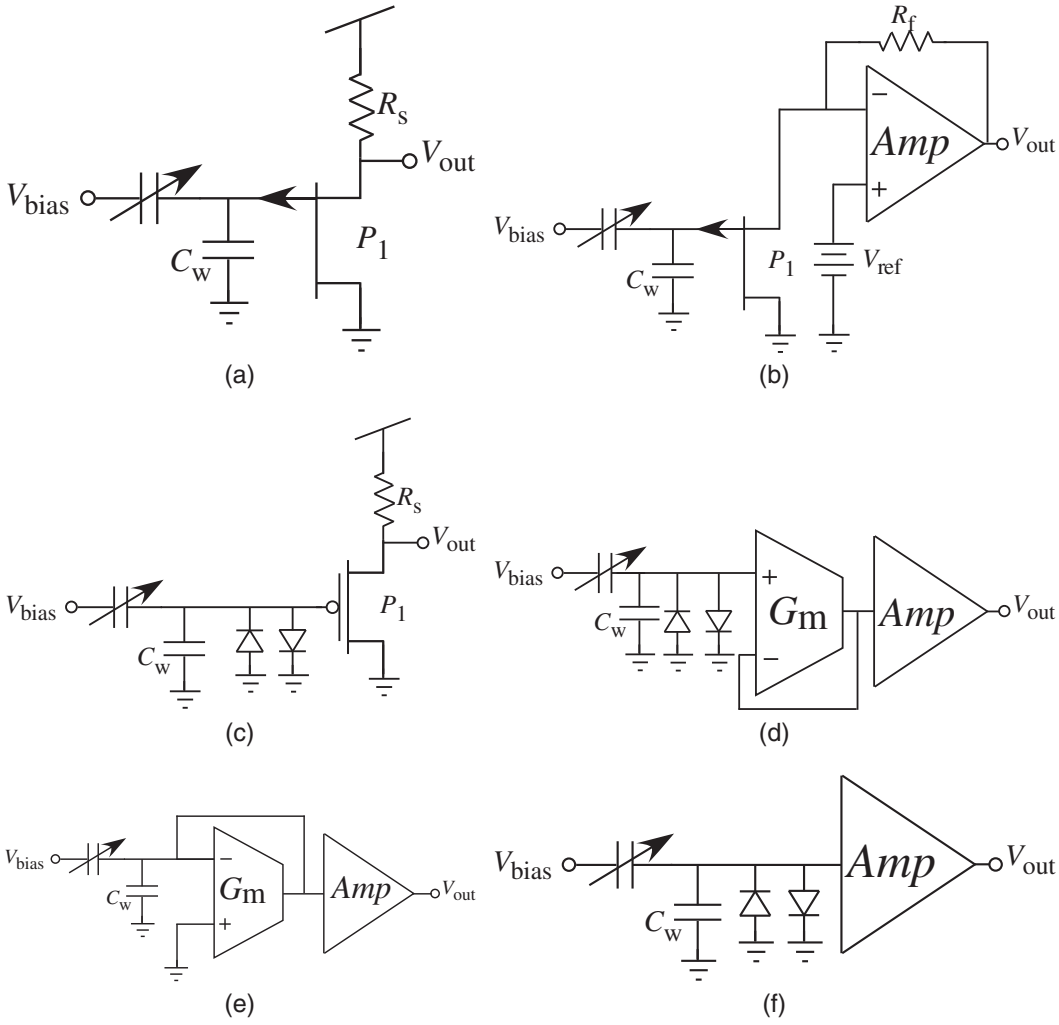


Figure 3.18. The block diagrams of the previous approaches to capacitive sensing for MEMS microphones.
(a) A self-biased JFET buffer as a microphone interface circuit
(b) The current through JFET is sensed and amplified to improve PSRR
(c)–(f) Diodes or linearized OTA are used as a large resistor and the voltage is directly amplified or buffered and then amplified.

CHAPTER 4

CAPACITIVE SENSING FOR MICROMACHINED ULTRASONIC TRANSDUCER

This chapter describes the use of a capacitive feedback charge amplifier as a receiver circuit for capacitive micromachined ultrasonic transducers (CMUTs). Compared with conventional approaches, using a charge amplifier, instead of a transimpedance amplifier, to detect capacitance variation avoids the dilemma of sensitivity-bandwidth tradeoff. A version of the capacitive sensing charge amplifier was fabricated in a 0.5 μm CMOS process. The chip contains a 8-to-1 multiplexer and is interfaced with a CMUT annular-ring array designed for forward-looking intravascular ultrasound imaging applications. Pulse-echo experiments were conducted in an oil bath using a planar target 3 mm away from the array and the measurement results show a signal-to-noise ratio of 16.65 dB with 122 μW power consumption around 3 MHz.

4.1 CMUTs for Intravascular Ultrasound Imaging

Currently, intravascular ultrasound (IVUS) has become a valuable diagnostic tool in the assessment of the extent of coronary artery disease, the leading cause of death in the United States. IVUS provides the unique possibility to image the arterial vessel wall *in vivo*, allowing one to study the coronary morphology during life and over time. Repeated IVUS investigations allow for tracking the natural course of atherosclerotic lesions and the effects of therapeutic measures. IVUS also plays an important role in the mechanistic assessment of treatments such as balloon angioplasty stent implantation and more recently drug eluting stents. New techniques require IVUS systems to generate images with higher resolution and higher frame rates and also to have sensitivity over a broad frequency range to allow harmonic imaging. In IVUS applications, piezoelectric transducer technology has prevented effective implementation of systems with diameters below 1mm. As CMUTs

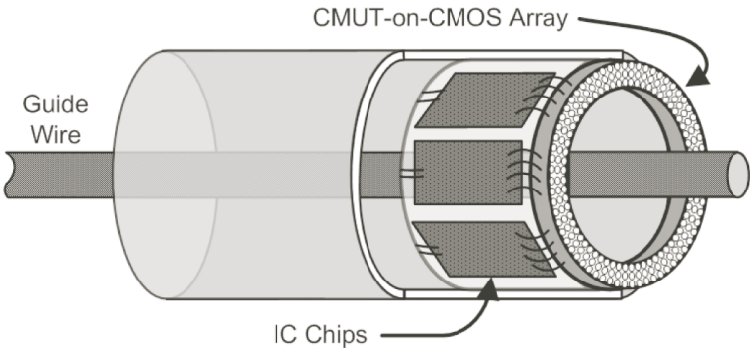


Figure 4.1. The block diagram of the IVUS catheter with integrated front-end electronics.

have advantages of wide bandwidth, ease of fabricating large arrays of practically any size and shape, and potential for integration with electronics, they have emerged as an attractive alternative to piezoelectric transducers for ultrasound imaging.

The latest advancements in the CMUT technology have enabled the construction of forward-looking (FL) annular-ring transducer arrays that can be placed in front of a catheter [27]. Because of the small size of each CMUT element in an FL-IVUS array, which is on the order of 100 μm and is much smaller than that in a non-invasive 1-D CMUT array, the parasitic capacitances introduced by the electrical interconnects can easily overwhelm the device capacitance and impair the achievable SNR. Either integrating the CMUT array with electronics on the die level or building the CMUT array directly on CMOS electronics, as illustrated in Figure 4.1, can avoid the performance degradation caused by the cable losses.

4.2 Capacitive Sensing for CMUTs

The design of the sensing circuits for CMUTs is challenging. The transducers usually operate at a high frequency because IVUS imaging requires a high resolution and because the imaging depth for vessels is relatively shallow. Hence, the sensing circuits need to have high bandwidth. Since the probe is located inside the patient's body, the power dissipation of the sensing circuits is another major concern. In brief, sensing a minute capacitance variation in the presence of large parasitic capacitances, and providing high bandwidth and

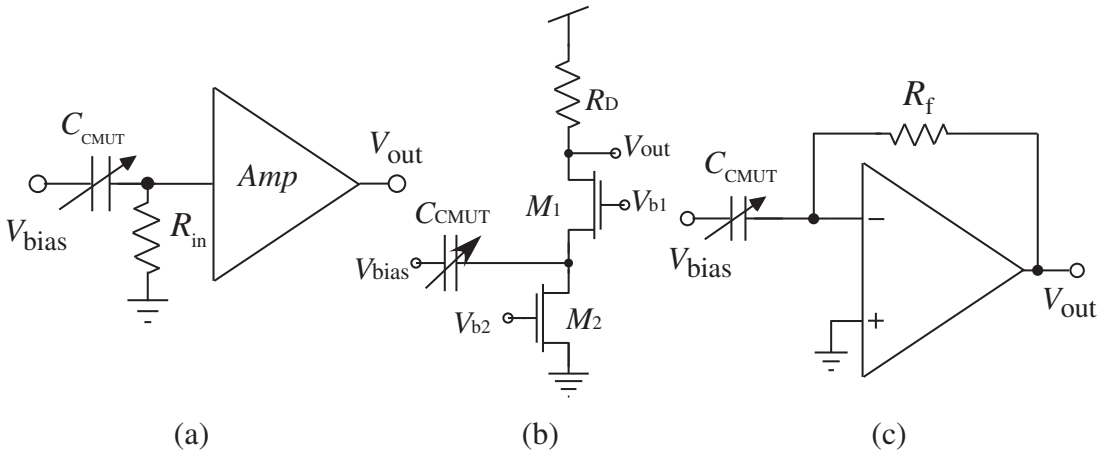


Figure 4.2. Conventional approaches to CMUT sensing.
(a) Resistive termination followed by an amplifier stage
(b) Common-gate amplifier as the input stage
(c) Resistive feedback transimpedance amplifier

a large dynamic range with low power dissipation pose significant challenges for the design of CMUT sensing circuits.

Conventionally, CMUT signals are converted from capacitive currents into voltages [28] using resistive terminations followed by amplifiers, using common-gate amplifiers, or using resistive feedback transimpedance amplifier (TIA), as shown in Figure 4.2. The first approach suffers from the direct tradeoff between the bandwidth and the input-referred current noise because they both are proportional to $1/R_{in}$. In the common-gate topology, although the noise can be minimized by maximizing the load resistance and the overdrive voltage of M_2 without affecting the bandwidth, this incurs a reduction in the output voltage headroom. In the last case, the feedback resistance does not limit the voltage headroom and the input capacitance can be reduced by the amplifier gain because of the shunt-shunt feedback. Therefore, the TIA topology is widely used in capacitive sensing applications. However, when the operating frequency is high, the bandwidth can be limited by the parasitic feedback capacitance.

The capacitive feedback charge amplifier with a charge adaptation circuit described in the previous chapter can also be used to sense CMUT signals. In the next section, the analysis of the charge amplifier with the adaptation circuit is presented and compared with

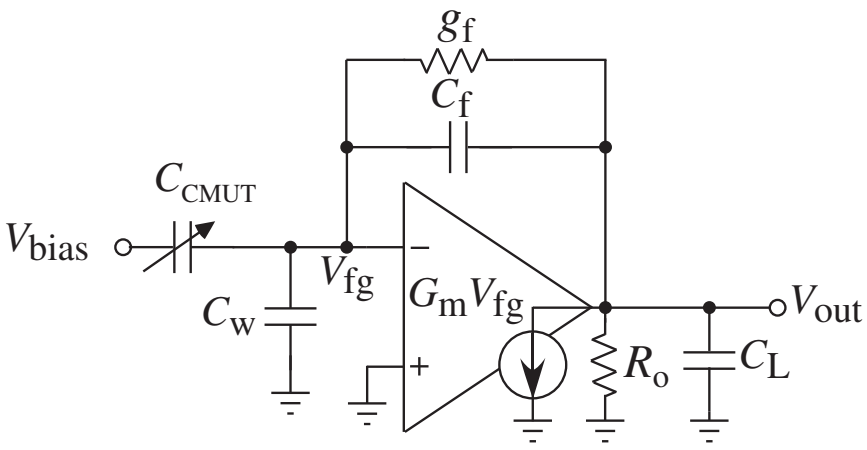


Figure 4.3. The small-signal model of a charge amplifier with charge adaptation feedback.

the transimpedance amplifier approach.

4.3 The Charge Amplifier vs. The Transimpedance Amplifier

The charge adaptation circuit in Figure 3.2 can be modeled by a small feedback conductance, g_f, as shown in Figure 4.3. The transfer function of the sensing circuit can be expressed as

$$\frac{V_{out}(s)}{C_{CMUT}(s)} = \frac{V_{bias}}{g_f} \cdot \frac{s(s\frac{C_f}{G_m} - 1)}{s^2\frac{C_T C_o - C_f^2}{G_m g_f} + s\left[\frac{C_f + C_T/A}{g_f} + \frac{C_T + C_o - 2C_f}{G_m}\right] + 1}, \tag{4.1}$$

where A is the amplifier gain. As shown in Figure 4.4, the adaptation scheme creates an extra zero at the origin and an extra low-frequency pole around g_f/C_f, assuming A is larger than C_T/C_f. If the transistors are in the subthreshold region, the minimum detectable capacitance can be expressed as

$$\Delta C_{min,CA} = \frac{1}{V_{bias}} \cdot \sqrt{\frac{nqU_T C_T C_f}{2\kappa C_o}}. \tag{4.2}$$

It is interesting to note that Figure 4.3 can also be viewed as a small signal model of a TIA with a parasitic feedback capacitance. The expression of (4.1) can be rearranged to

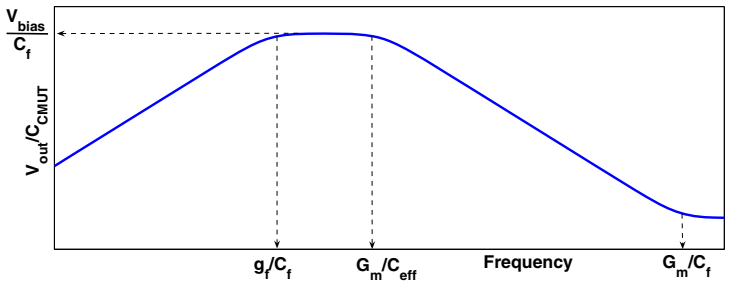


Figure 4.4. Simulated frequency response of the capacitive feedback charge amplifier with a feedback resistance.

describe the transfer function of the TIA as

$$\frac{V_{\text{out}}(s)}{I_{\text{CMUT}}(s)} = \frac{V_{\text{out}}(s)}{sV_{\text{bias}}C_{\text{CMUT}}(s)} = \frac{1}{g_{\text{f}}} \cdot \frac{s\frac{C_{\text{f}}}{G_{\text{m}}} - 1}{s^2\frac{C_{\text{T}}C_{\text{o}}-C_{\text{f}}^2}{G_{\text{m}}g_{\text{f}}} + s\left[\frac{C_{\text{f}}+C_{\text{T}}/A}{g_{\text{f}}} + \frac{C_{\text{T}}+C_{\text{o}}-2C_{\text{f}}}{G_{\text{m}}}\right] + 1}. \tag{4.3}$$

The minimum detectable capacitance of the TIA can be derived as

$$\Delta C_{\text{min,TIA}} = \frac{g_{\text{f}}}{\omega_0 V_{\text{bias}}} \cdot \sqrt{\frac{nqU_{\text{T}}C_{\text{T}}}{2\kappa C_{\text{f}}C_{\text{o}}}}, \tag{4.4}$$

where ω_0 is the operating frequency.

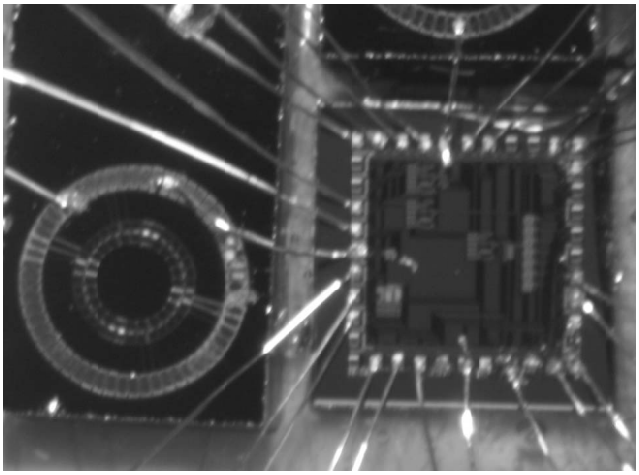
Although the topologies of a TIA and a charge amplifier are the same, their design philosophies are different. In a typical TIA design, the operating frequency should be lower than the first pole. Therefore, the operation region is the ascendent region in Figure 4.4. The sensitivity-bandwidth trade-off of a TIA is obvious from (4.3) and (4.4). Increasing the bandwidth by increasing g_{f} results in decreased sensitivity. On the other hand, using a charge amplifier to sense CMUT signals can avoid all the dilemmas mentioned before. The sensitivity can be improved by choosing large values of V_{bias} and C_{L} and a small value of C_{f}. The bandwidth corresponding to the second pole can be extended by using a larger value of G_{m}.

4.4 Pulse-Echo Measurement

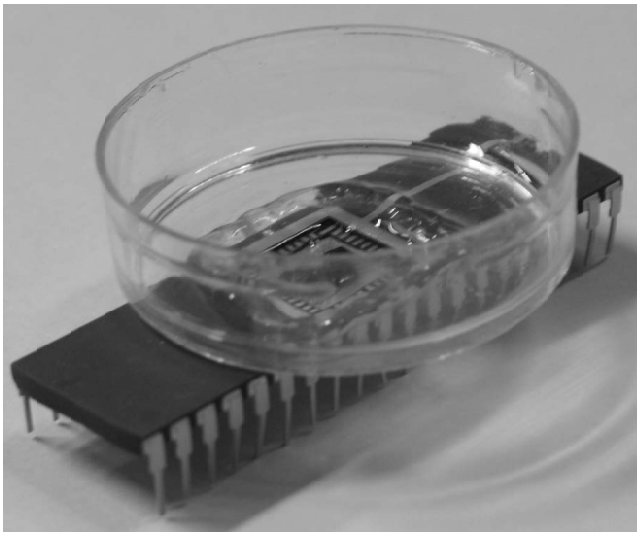
A version of the charge amplifier that uses a *p*MOS transistor as the charge adaptation feedback was fabricated. As shown in Figure 4.5a, the chip with electronics is wire bonded to an annular-ring IVUS CMUT array [29]. The size of each element is $70\,\mu\text{m} \times 70\,\mu\text{m}$,

giving rise to a measured capacitance of 2 pF. A petri dish with an opening at the bottom is glued on top of the package using epoxy. During measurement, transducers and the circuit are immersed in a vegetable oil bath, as shown in Figure 4.5b.

By applying different bias voltages to the feedback transistor, the same circuit can be configured either as a TIA or as a charge amplifier. Because the charge effect resulting from the capacitance change is equivalent to that resulting from the voltage change, the frequency response of the circuit can be measured by applying an ac signal at one of the CMUT terminals. Measured results are shown in Figure 4.6. As the operating frequency is



(a)



(b)

Figure 4.5. Photographs of the setup for the testing of CMUT sensing.

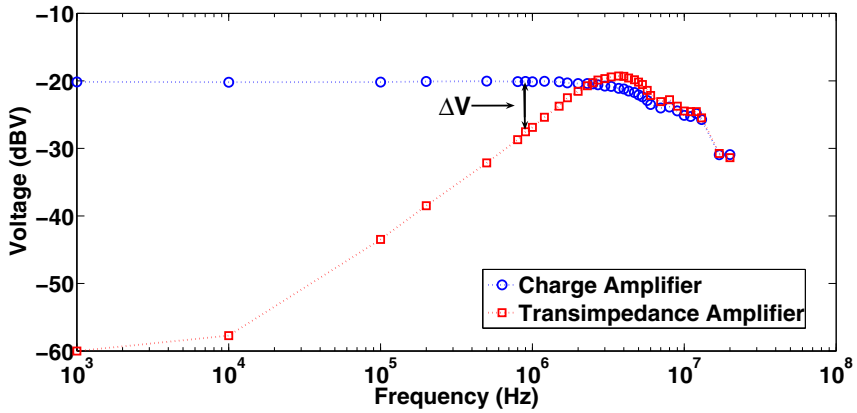


Figure 4.6. Comparison of the frequency response of a charge amplifier and a transimpedance amplifier.

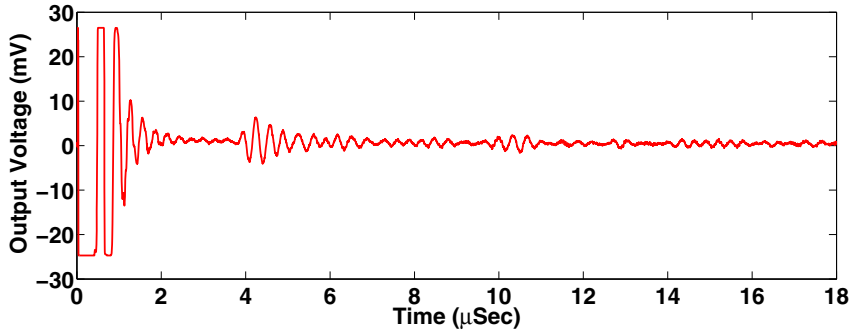


Figure 4.7. Measured pulse-echo response from a charge amplifier.

below the second pole, the charge amplifier approach can generate larger output magnitude than the TIA approach given the same voltage (or capacitance) variation.

Pulse-echo experiments were performed using one CMUT device as a transmitter and the other element bonded to the circuit as a receiver. The transmitting element is stimulated by a 20 V-peak pulse. The receiving device is biased by a 70 V dc voltage and the extracted feedback capacitance is 200 fF. Both devices are immersed in the oil. The distance between these two devices is about 6 mm, corresponding to a pulse-echo distance from a planar target 3 mm away. The recorded waveform, shown in Figure 4.7, indicates a center frequency of 3 MHz, which is mainly limited by the amplifier bandwidth. The measured output noise floor is 2.5 mVrms and the measured SNR from the first received acoustic signal is 16.65 dB. The power consumption of the charge amplifier is only 122 μW.

4.5 Discussion

For the charge amplifier approach, the bandwidth of the circuit is determined by the second pole of (4.1) that is $(G_{\mathrm{m}}C_{\mathrm{f}})/(C_{\mathrm{T}}C_{\mathrm{o}})$. By contrast, the bandwidth of the transimpedance amplifier is set by the first pole of (4.1) that is $g_{\mathrm{f}}/C_{\mathrm{f}}$. If the bandwidth of the first pole is pushed toward higher frequencies and is laid over the second pole, and if there is some margin between the operating frequency and the second pole, then the output from the transimpedance amplifier is smaller than the output from the charge amplifier. The difference of input referred current noise between these two approaches results from the different values of the feedback resistance. Because the feedback resistance used in the charge amplifier is higher than that used in a TIA, the corresponding input referred current noise is smaller and the SNR is higher compared to the TIA approach. However, if the sensor noise is larger than the circuit input referred noise, the SNR advantage of the charge amplifier approach is smeared.

Another difference between these two approaches is the information represented by the output voltages. For a charge amplifier, the output voltage is proportional to the charge that corresponds to the displacement of the membrane. For a transimpedance amplifier, the output voltage is proportional to the current that corresponds to the velocity of the moving membrane. In the case of MEMS microphone operating in air, the acoustic impedance of the transducer is small and the pressure is proportional to the displacement of the membrane. The charge amplifier provides the information of pressure directly. For the CMUT in oil or water, the acoustic impedance is around 1.5×10^6 Rayl, 10,000 times larger than that in air. In this case, the pressure is proportional to the membrane velocity. Therefore, a differentiator following the charge amplifier is needed to convert the charge information to the current information. In this case, the noise contributed by the differentiator needs to be considered in performance evaluation.

CHAPTER 5

RECONFIGURABLE ANALOG SIGNAL PROCESSOR WITH UNIVERSAL SENSOR INTERFACE

The field-programmable analog array (FPAA) is a powerful tool for fast prototyping analog systems. Innovative design ideas can be quickly realized and tested in hardware without time-consuming and expensive silicon fabrication. In advanced FPAAs, floating-gate transistors have been used as switches in the interconnect network and as programmable analog elements resulting in a compact reconfigurable analog signal processor. This chapter presents an updated version of the floating-gate based FPAA with universal sensor interface blocks. This reconfigurable analog signal processor can thus interface with different sensors and perform analog signal processing algorithms.

5.1 Floating-Gate Based FPAA

Different analog circuits for different applications can be synthesized and tested on an FPAA. The architecture of a generic FPAA is shown in Figure 5.1. The reconfigurability is mainly achieved through the interconnect network that is composed of an array of switches and memory cells. The programmability of the traditional FPAA mainly comes from the digital-to-analog converters or from ratios of standard components that consume a huge amount of silicon area.

The ability to precisely program floating-gate transistors makes it possible to include more analog programmable elements in an FPAA [1]. For example, if transistors are in weak inversion, a floating-gate based current mirror shown in Figure 5.2a provides the same function as the programmable element shown in Figure 5.1 and consumes much less area. A floating-gate transistor shown in Figure 5.2b is the compact integration of a memory cell and a switch. Therefore, the FPAA interconnect network can be made from a floating-gate transistor array. Measured I-V characteristics of a floating-gate switch and the comparisons

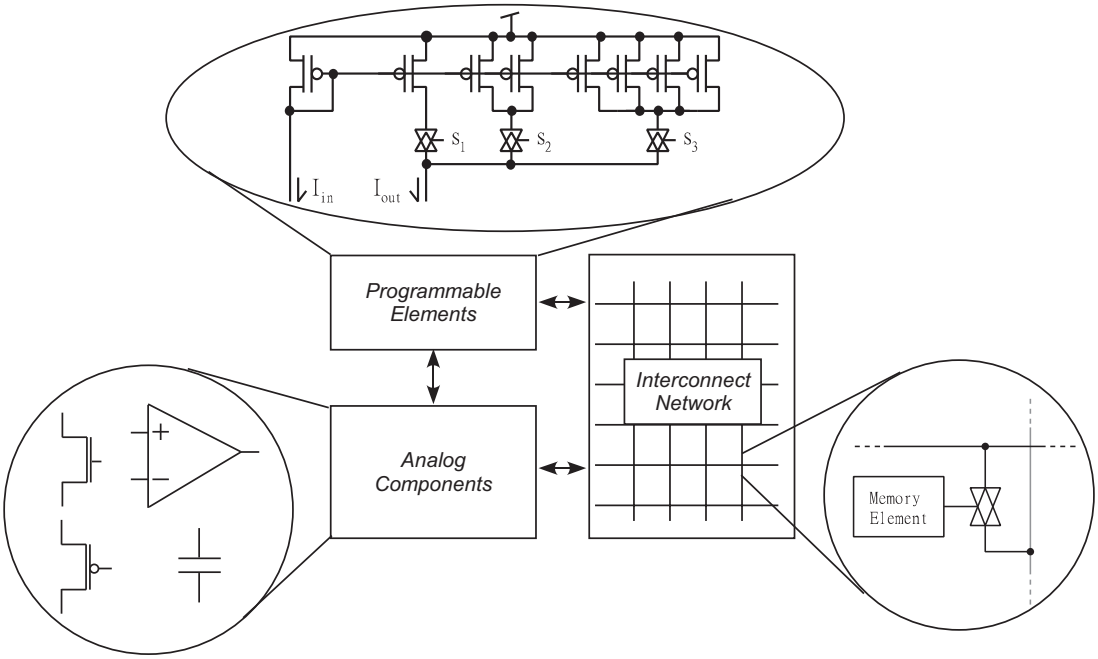


Figure 5.1. The architecture of a field programmable analog array.

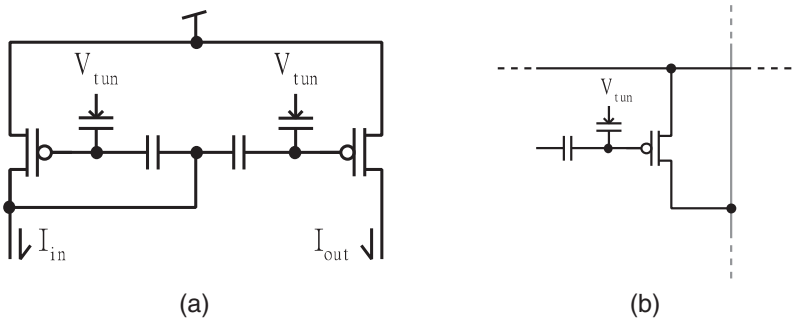


Figure 5.2. Floating-gate based programmable elements and switch for FPAA [1].
(a) A floating-gate based current mirror
(b) A floating-gate based switch element

of the on-resistance are shown in Figure 5.3. The floating-gate switch is highly compact and has low parasitic resistance that is comparable to a transmission gate switch. Several versions of floating-gate based large-scale FPAA have been developed and tested [1].

5.2 RASP 2.8 Design Overview

The device RASP 2.8a presented in this section is a vanilla version of a large-scale floating-gate based FPAA fabricated in a 0.35 μm CMOS process. It is composed of 32 configurable

analog blocks (CABs) and a three-level routing network. This device contains over 50,000 programmable analog elements and an on-chip programming circuitry that can program more than 200 floating-gate transistors per second.

The block diagram of the RASP 2.8a is shown in Figure 5.4. Two types of CABs are arranged into a 8×4 array and are connected through local and global switch matrices. Components in the first type of the CABs include three operational transconductance amplifiers (OTAs), a voltage buffer, three floating capacitors (500 fF each), and nMOS/pMOS transistor arrays with two common terminals for easily constructing source-follower or current-mirror topologies. All the OTAs are biased using floating-gate transistors to provide the option for bandwidth, noise and power tradeoffs. Two of the OTAs and one output buffer have floating-gate differential pairs so that the offset of the amplifier can be programmed. Because of the capacitive divider at the input stage, these elements have a wide input linear range that is essential to reduce the distortion in Gm-C filters and oscillators. The components in the second type of the CABs include two wide-linear-range folded Gilbert multipliers, a wide-linear-range OTA, and a programmable current mirror. These CAB components are connected through the floating-gate switch-matrices that can also be used as analog computation primitives.

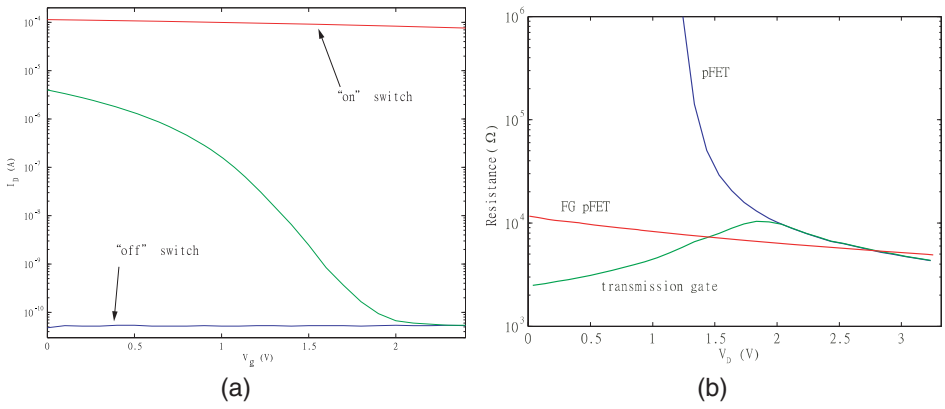


Figure 5.3. The characteristics of a floating-gate switch for FPAAs [1].
(a) The I-V curves of a floating-gate switch
(b) Measured Resistance of a floating-gate switch

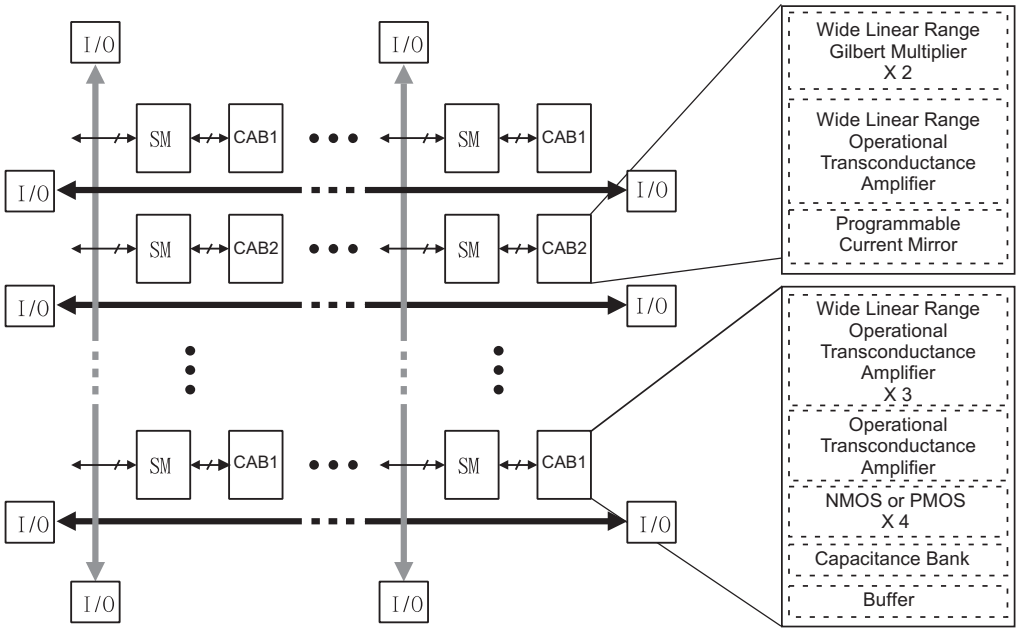


Figure 5.4. The block diagram of the RASP 2.8a.

The routing architecture of the RASP 2.8x series is shown in Figure 5.5 to illustrate the three-level interconnection, local, nearest neighbor, and global. High speed circuits can be synthesized using the local or the nearest neighbor connections to minimize the parasitic capacitance. The global connections are used for the input and output interface. A voltage buffer is used to isolate the pad capacitance from the CAB components. Thus routing between CABs can be accomplished with relatively lower parasitic and can achieve bandwidths of approximately 6 MHz at around 100 nA of current. The achievable bandwidth within a CAB should be an order of magnitude higher. The other feature of the routing scheme is the bridge transistors. They allow local lines to be bridged between CABs facilitating variable length connections without incurring the capacitance penalty of global lines.

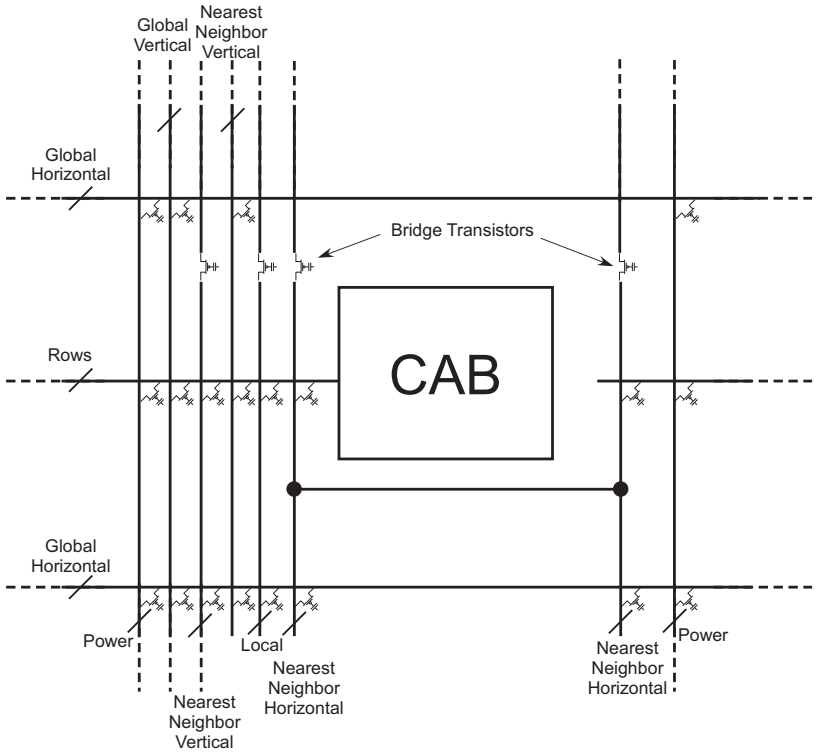


Figure 5.5. The architecture of routing in the RASP 2.8x series.

5.3 FPAA With Universal Sensor Interface

In this research, a universal sensor interface (USI) block is designed for the RASP 2.8c, a version of the large-scale floating-gate based FPAA for sensor applications. The architecture of the RASP 2.8c is shown in Figure 5.6. There are 8 USI blocks located at the top of the chip to interface with sensors. The received signals are then sent to the subsequent CABs for further signal processing. The layout and the micrograph of the RASP 2.8c are shown in Figure 5.7.

The schematics of components in a USI block and the interconnect network are shown in Figure 5.8. To reduce the parasitic capacitance at the interface between the sensor and the FPAA, the floating-gate switches are not exhaustively spread in the interconnect network. Only the interconnects with circles shown in Figure 5.8 exist floating-gate switches. The components in a USI block include a self-biased cascoded common-source amplifier, a 9-transistor OTA, an operational amplifier with an output stage, a transmission gate for

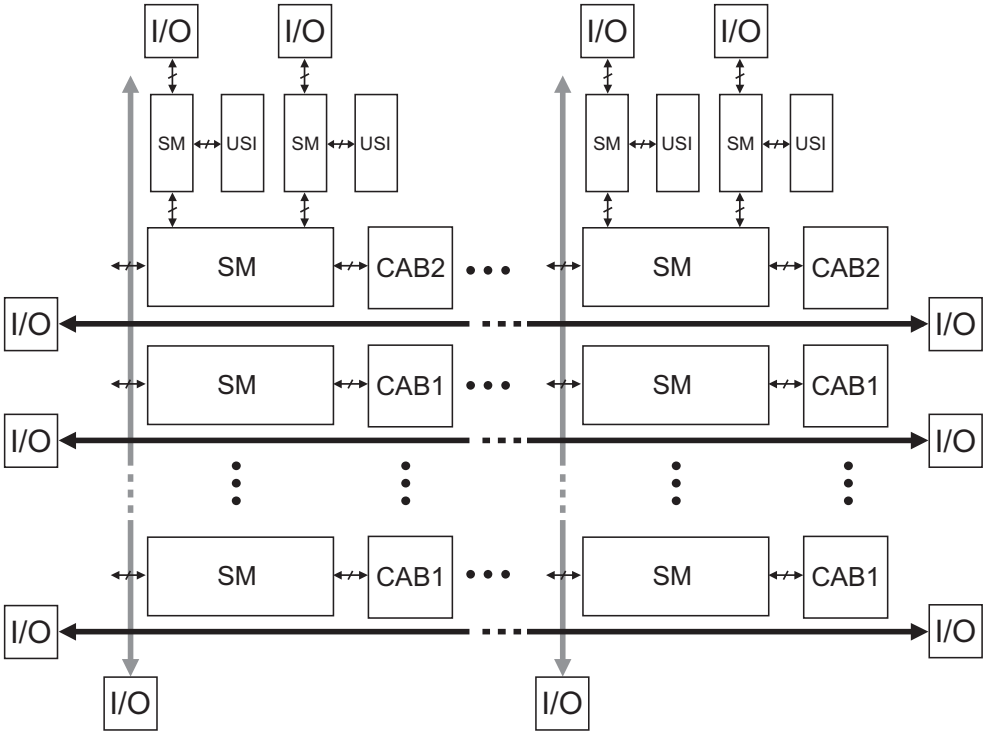


Figure 5.6. The architecture RASP2.8C.

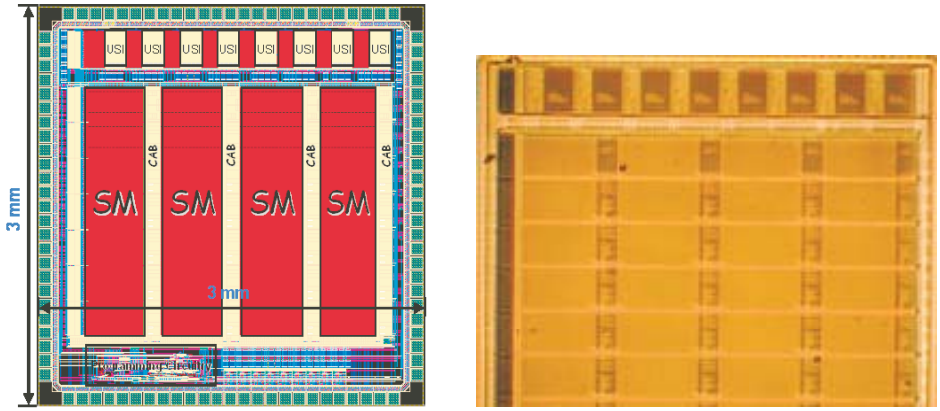


Figure 5.7. The layout and the micrograph of the RASP 2.8c.
(a) The layout of the RASP 2.8c
(b) The micrograph of the RASP 2.8c

multiplexing, a source follower, two sets of capacitor bank, two poly-resistors, two MOS-BJT nonlinear resistors and a floating-gate pMOS transistor. The interface circuit in the RASP 2.8c can be synthesized for capacitive sensing, current sensing, or voltage sensing

using the components inside a USI block. In Figure 5.9, a capacitive sensing charge amplifier is synthesized in a USI block by turning on proper switches. Measured frequency response is shown in Figure 5.10.

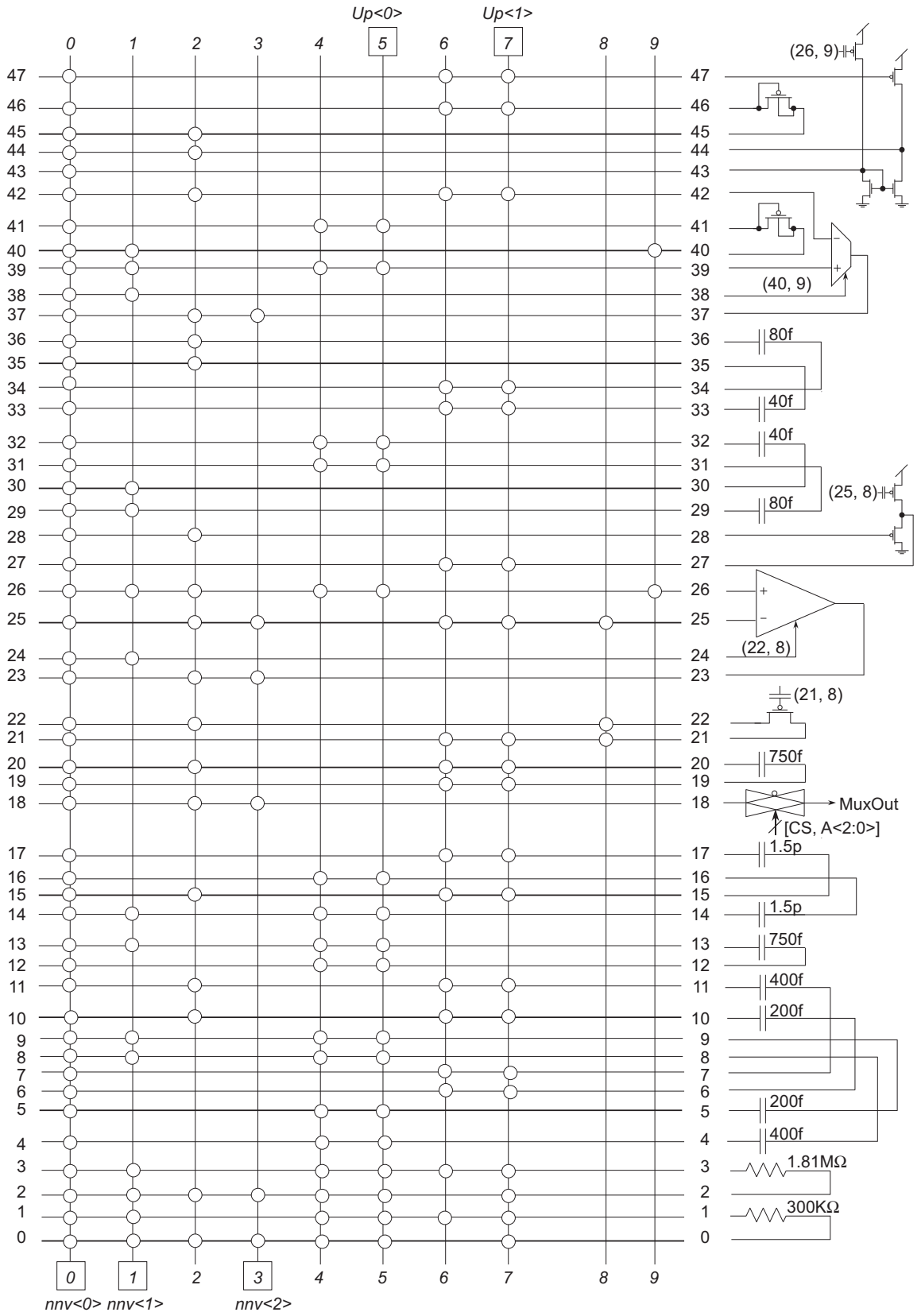


Figure 5.8. The fuse-network and components of a USI.

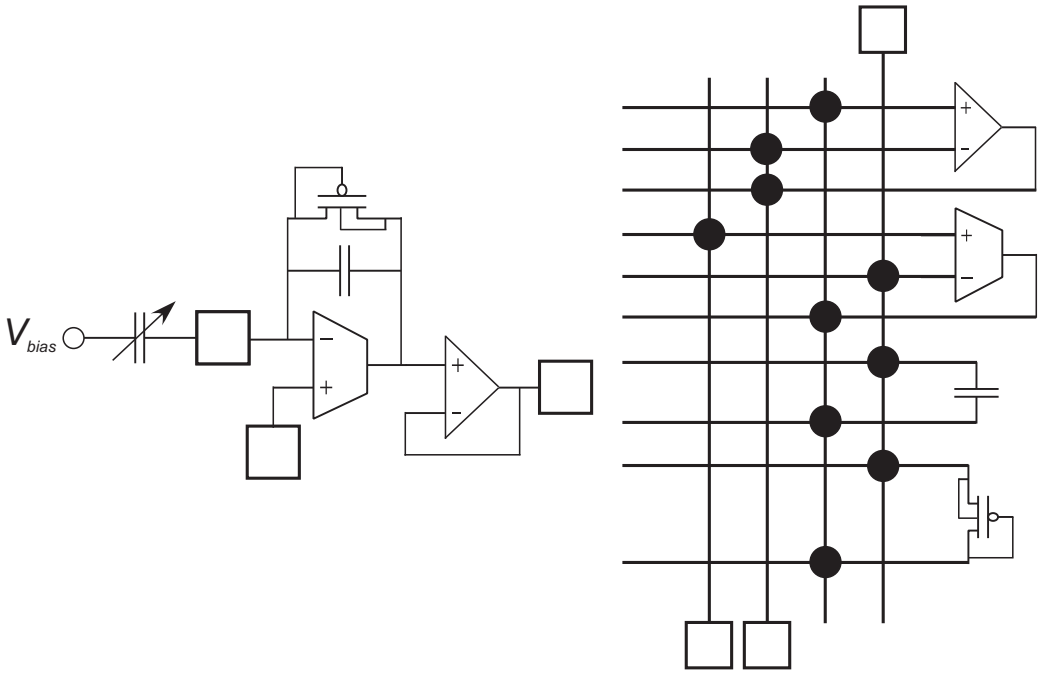


Figure 5.9. A capacitive sensing charge amplifier synthesized in the RASP 2.8c.

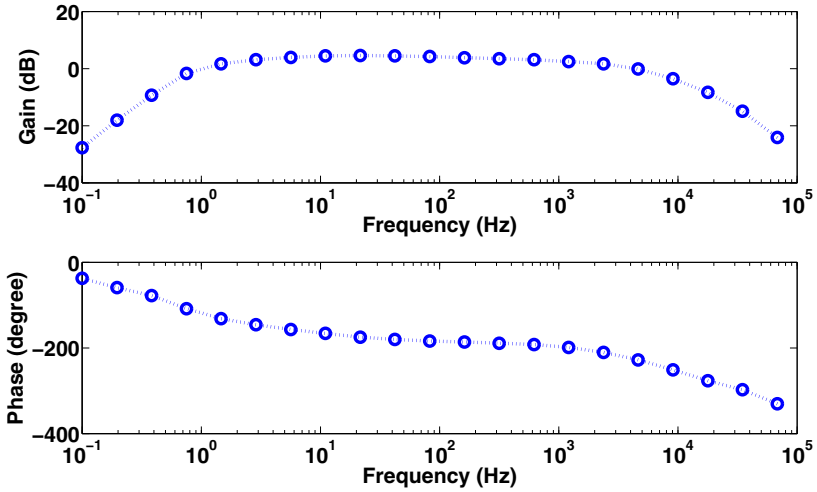


Figure 5.10. The frequency response measured from a charge amplifier synthesized in the RASP 2.8c.

CHAPTER 6

ANALOG IMPLEMENTATION OF RADIAL BASIS FUNCTIONS

Radial basis functions (RBFs) are widely used as similarity measures in many recognition and classification applications. To efficiently realize Gaussian or Gaussian-like radial basis functions in analog neural networks or classifiers, many analog RBF circuits have been proposed [30–35]. Among these previous works, the "bump" circuit in [30] is the most classic because of its simplicity. However, the width of the transfer curve in the conventional bump circuit is not adjustable. Another drawback of these previous works is the requirement of extra hardware to store or to periodically refresh template data when they are employed in recognition systems. In the works of [13, 36, 37], template data are stored as charges on floating-gate transistors that are utilized to implement a bump circuit. These floating-gate approaches result in very compact analog classifiers. However, widths of the transfer curves in these floating-gate circuits are still fixed.

In this chapter, the classic simple bump circuit and some floating-gate bump circuits are introduced first. A detailed description of an improved fully-programmable floating-gate bump circuit that has been designed, tested, and successfully employed in some large-scale analog classifiers is presented. Measurement results from a prototype chip are provided in this chapter as well. The height, the width, and the center of the circuit's transfer curve are mathematically related to the maximum likelihood, the variance, and the mean of a distribution, respectively. With the ability to individually program these three parameters, the resultant classifiers can fit into different scenarios and can use all of the statistical information up to the second moment.

6.1 Conventional Bump circuit

The bump circuit proposed in [30] is a small analog circuit for computing the similarity of two voltage inputs. The output current from the circuit becomes large when the input voltages are close to each other and decreases exponentially when the input voltage difference increases.

The schematic of this bump circuit is shown in Figure 6.1a. If all transistors operate in the subthreshold region, the branch currents in the differential pair can be expressed as

$$I_1 = \frac{I_b}{1 + e^{-\kappa \Delta V_{in}/U_T}} \tag{6.1}$$

and

$$I_2 = \frac{I_b}{1 + e^{\kappa \Delta V_{in}/U_T}}, \tag{6.2}$$

where κ is the subthreshold slope factor, U_T is the thermal voltage, and $\Delta V_{in} = V_{in1} - V_{in2}$. Transistors M_3 to M_6 form a current correlator that computes the harmonic mean of I_1 and I_2. If the aspect ratio of transistors M_5 and M_6 is w times larger than that of M_3 and M_4, the

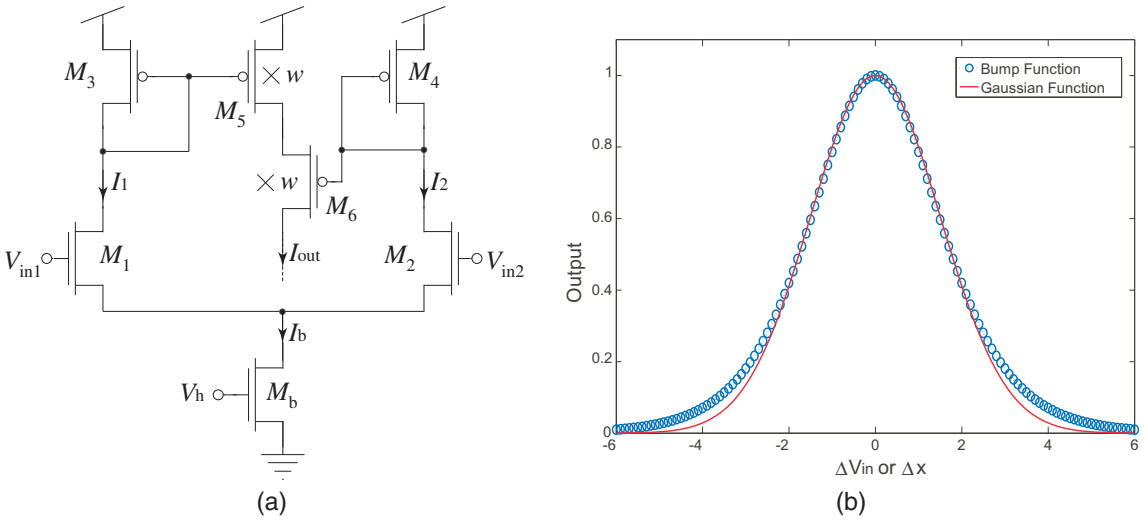


Figure 6.1. A simple bump circuit and its Gaussian-like DC transfer curve.
(a) The schematic of the simple bump circuit in [30]
(b) The DC transfer curve of the bump circuit compared with a normalized Gaussian function

output current can be described as

$$\begin{aligned} I_{\text{out}} &= \frac{wI_1I_2}{I_1+I_2} \\ &= \frac{wI_{\text{b}}}{2+e^{\kappa\Delta V_{\text{in}}/U_{\text{T}}}+e^{-\kappa\Delta V_{\text{in}}/U_{\text{T}}}} \\ &= \frac{wI_{\text{b}}}{2}\text{sech}^2\left(\frac{\kappa\Delta V_{\text{in}}}{2U_{\text{T}}}\right). \end{aligned} \tag{6.3}$$

The output current reaches a maximum value when two input voltages are equal. The DC transfer curve is shaped like a Gaussian function; therefore, this simple circuit can be used to approximate a Gaussian function. For comparison purposes, a plot that overlays the normalized bump circuit transfer function and the normalized Gaussian function is shown in Figure 6.1b.

A disadvantage of this circuit is that the width of the DC transfer characteristic is fixed by the ratio of U_{T}/κ, which is not adjustable. To overcome this drawback and to provide a DC transfer function with a tunable width, some previous designs have used differently sized capacitors [35] or transistors [31] to be switched into their circuits, or have used various techniques to change the current flow of a differential pair [32,33], which alters the transconductance, in their circuits. However, such modifications are usually complicated and are not intrinsically programmable.

6.2 Previous Works on Floating-Gate Bump Circuits

The aforementioned analog RBF approaches have another shortcoming in implementing analog classifiers. Extra hardware is required to store or to periodically refresh the template data. Since floating-gate transistors can be used as analog memories to store the template information, they have been employed to implement the bump circuits that lead to the design of compact classification systems [13, 36, 37].

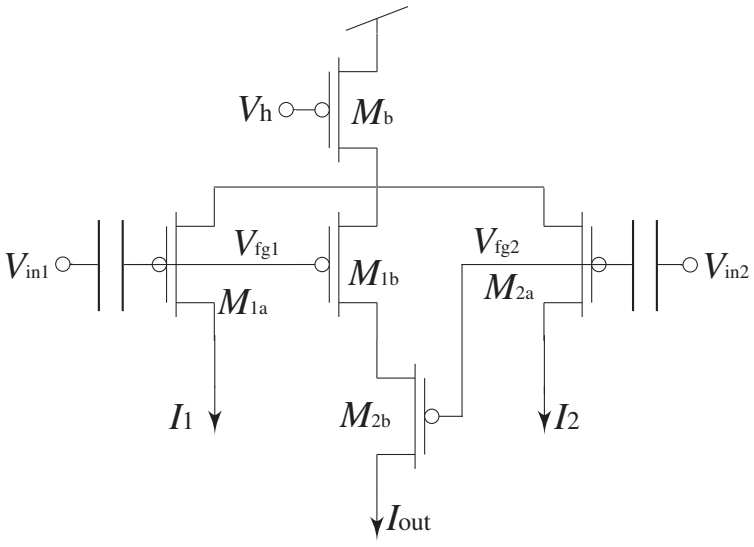


Figure 6.2. The schematic of a compact floating-gate bump circuit.

The schematic of a compact floating-gate bump circuit [13, 36, 37] is shown in Figure 6.2. The output current can be expressed as

$$\begin{aligned} I_{\text{out}} &= \frac{I_{\text{b}}}{2}\text{sech}^2\left(\frac{\kappa(V_{\text{fg1}} - V_{\text{fg2}})}{2U_{\text{T}}}\right) \\ &\approx \frac{I_{\text{b}}}{2}\text{sech}^2\left(\frac{\kappa(\Delta V_{\text{in}} - \Delta V_{\text{Q}})}{2U_{\text{T}}}\right), \end{aligned} \tag{6.4}$$

where $\Delta V_{\text{Q}} = (Q_2 - Q_1)/C_{\text{T}}$, Q_1 and Q_2 are the charges at floating nodes, and C_{T} is the total capacitance seen from a floating gate. The input signal, ΔV_{in}, is compared with the template data, ΔV_{Q}. and the output current is proportional to the similarity between the input and the template. If the charges at the floating nodes are adapted according to the inputs, then this circuit can be used in an adaptive system performing competitive learning algorithms [13,37]. Although the center of the transfer curves in this circuit can be adapted to the mean value of the input distribution, the width remains constant.

The first version of the proposed floating-gate bump circuit that has a programmable transfer curve is shown in Figure 6.3a. A folded differential pair [38] is used to cancel the input common-mode voltage. The width of the transfer characteristic depends on V_{width} and the charges stored on the floating-gates transistors M_3 and M_4. The center is controlled by the differential charge at the floating-gate transistors $M_{1a,b}$ and $M_{2a,b}$. The height of the

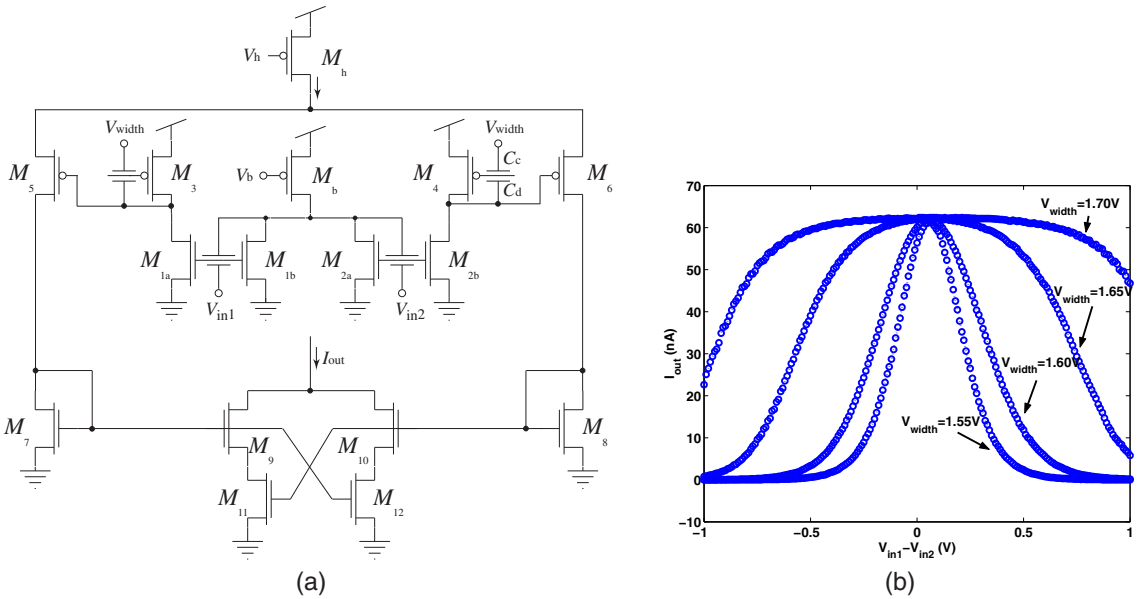


Figure 6.3. A proposed floating-gate bump circuit and the measured DC transfer curves.
(a) The schematic of a width-tunable floating-gate bump circuit
(b) The measured transfer characteristics

transfer curve is determined by the tail current of a differential pair. Measurement results are shown in Figure 6.3b. The input voltage range of this circuit is rail to rail. Although the width of the first version of the floating-gate bump circuit has a wide tunable range, the transfer curve is more like a trapezoid than like an exponentially decayed Gaussian curve when its width is large. To better approximate the Gaussian function, an improved version of the floating-gate bump circuit has been proposed and tested. It is described and analyzed in detail in the next section.

6.3 Improved Programmable Floating-gate Bump circuit

The schematics of an improved floating-gate bump circuit and its bias generation block are shown in Figure 6.4. All floating-gate transistors have two input capacitances and all input capacitances are of the same size. If all of the parasitic capacitances as well as the tunneling capacitance are neglected, the floating-gate voltage can be expressed as

$$V_{\mathrm{fg}} = \frac{1}{2}(V_{\mathrm{con1}} + V_{\mathrm{con2}}) + V_{\mathrm{Q}}, \tag{6.5}$$

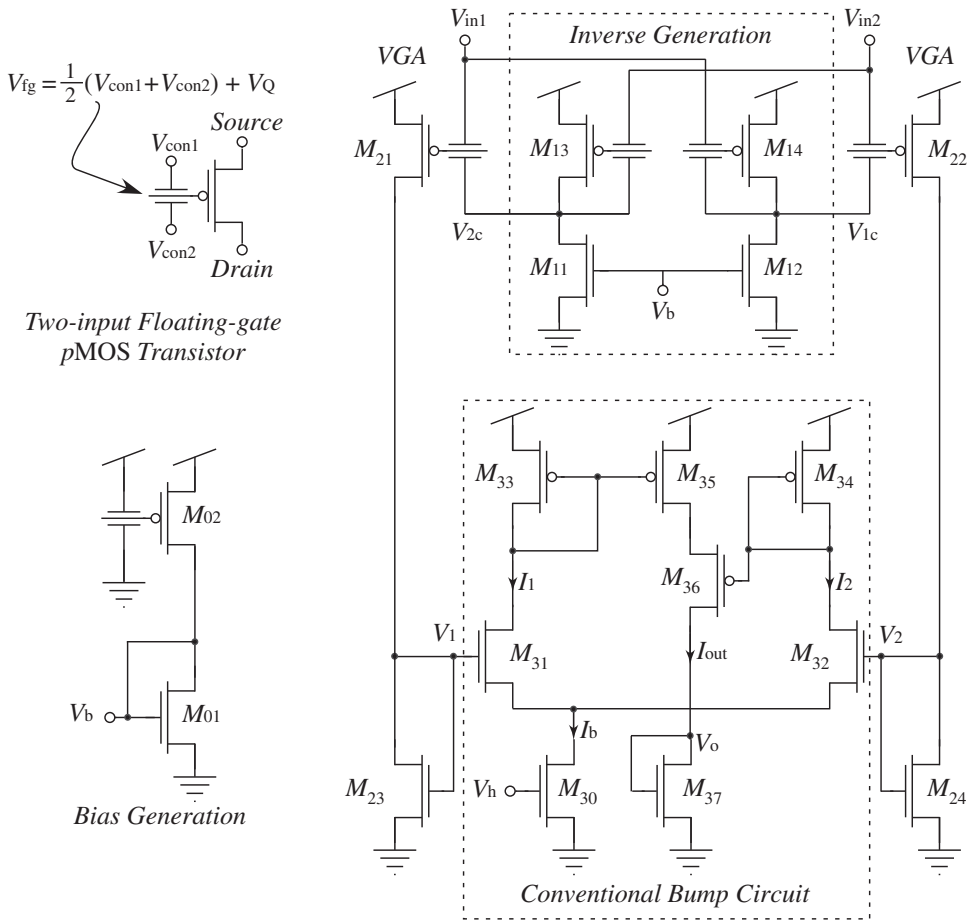


Figure 6.4. An improved version of the floating-gate bump circuit and its bias generation block.

where V_{con1} and V_{con2} are voltages applied to the control gates, $V_Q = Q_{fg}/C_T$, Q_{fg} is the charge on the floating gate, and C_T is the total capacitance seen from the floating gate.

This improved bump circuit is composed of three parts: an inverse generation block, a conventional bump circuit, and a fully-differential variable gain amplifier (VGA). The inverse generation block provides the complementary input voltages to the VGA so that the floating-gate common-mode voltage of M_{21} and M_{22}, as well as the outputs of the VGA, is independent of the input signal common-mode level. The width of the bell-shaped transfer curve can be adjusted by changing the VGA gain.

The inverse generation block is composed of two floating-gate summing amplifiers that generate complementary input voltages. If the charges on M_{13} and M_{14} are matched and

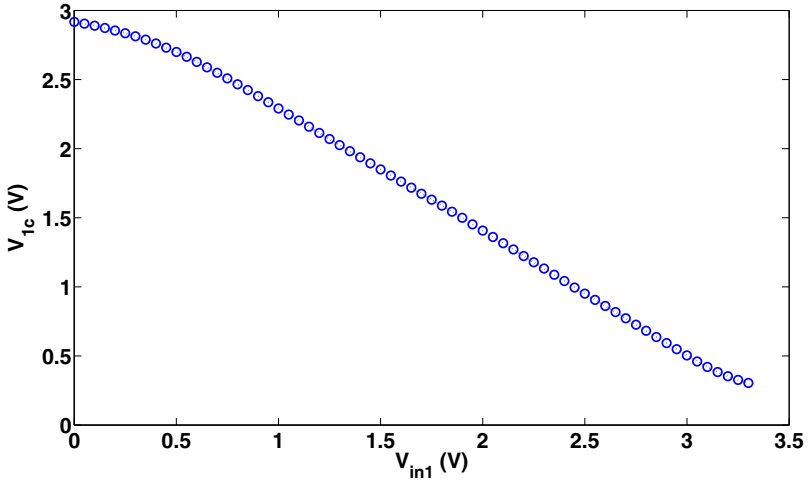


Figure 6.5. Transfer characteristics of the inverse generation block.

the transistors are in the saturation region, then the equality

$$V_{\text{in1}} + V_{\text{1c}} = V_{\text{in2}} + V_{\text{2c}} = V_{\text{const}} \tag{6.6}$$

is true, where V_{const} is only dependent on the bias voltage, V_{b}, and the charges on M_{13} and M_{14}. If the charge on M_{02} in the bias generation circuit also matches charges on M_{13} and M_{14}, the operating range of the summing amplifier, which is determined by the generated voltage V_{b}, is one V_{DSsat} away from the supply rails. The transfer characteristics of the inverse generation block are shown in Figure 6.5. The outputs of the summing amplifiers are fed to floating-gate transistors in the VGA and the outputs of the VGA are independent of the input common-mode signal.

To illustrate the operation of the inverse generation block, the floating-gate voltages of M_{21} and M_{22} are expressed as

$$\begin{aligned} \text{V}_{\text{fg,21}} &= \frac{1}{2}(V_{\text{in1}} + V_{\text{const}} - V_{\text{in2}}) + \frac{Q_{21}}{C_{\text{T}}} \\ &= \frac{1}{2}\Delta V_{\text{in}} + V_{\text{Q,cm}} + \frac{1}{2}V_{\text{Q,dm}} \end{aligned} \tag{6.7}$$

$$\begin{aligned} \text{V}_{\text{fg,22}} &= \frac{1}{2}(V_{\text{in2}} + V_{\text{const}} - V_{\text{in1}}) + \frac{Q_{22}}{C_{\text{T}}} \\ &= -\frac{1}{2}\Delta V_{\text{in}} + V_{\text{Q,cm}} - \frac{1}{2}V_{\text{Q,dm}}, \end{aligned} \tag{6.8}$$

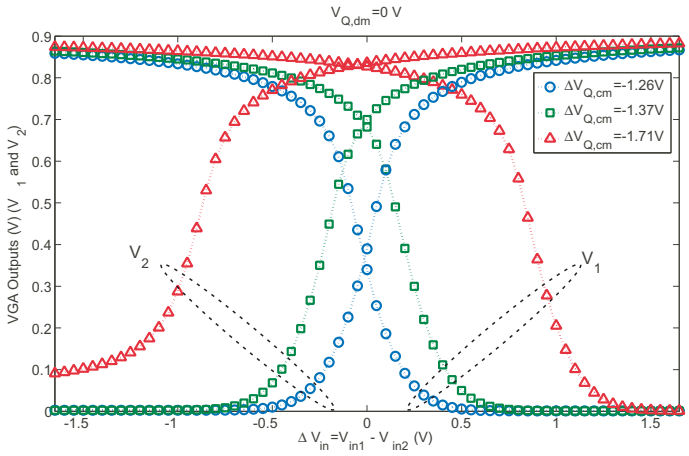
where $\Delta V_{\text{in}} = V_{\text{in1}} - V_{\text{in2}}$, Q_{21} and Q_{22} are the amounts of charge on M_{21} and M_{22} respectively, C_{T} is the total capacitance seen from a floating gate, and

$$V_{\text{Q,cm}} = \frac{1}{2}\left(\frac{Q_{21} + Q_{22}}{C_{\text{T}}} + V_{\text{const}}\right) \tag{6.9}$$

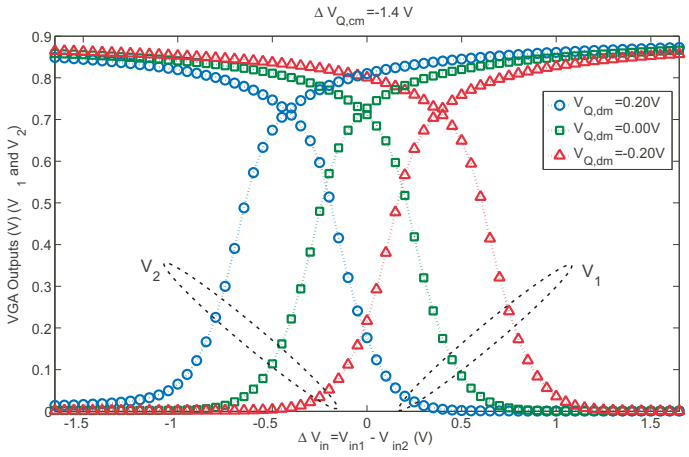
$$V_{\text{Q,dm}} = \frac{Q_{21} - Q_{22}}{C_{\text{T}}}. \tag{6.10}$$

From (6.7) and (6.8), these two floating-gate voltages are not dependent on the input signal common-mode level.

The variable gain of the VGA is derived from the nonlinearity of the transfer function



(a)



(b)

Figure 6.6. The transfer characteristics of the variable gain amplifier.
(a) The effect of the common-mode charge on M_{21} and M_{22}
(b) The effect of the differential charge on M_{21} and M_{22}

from the floating-gate voltage, $V_{\mathrm{fg},21}$ (or $V_{\mathrm{fg},22}$), to the diode-connected transistor drain voltage, V_1 (or V_2). Several pairs of the transfer curves corresponding to different amounts of charge on the floating gates are measured and are shown in Figure 6.6. The measurement is taken with $V_{\mathrm{in}2}$ fixed at $V_{\mathrm{DD}}/2$ while $V_{\mathrm{in}1}$ is swept from 0V to V_{DD}. The value of ΔV_{in} at the intersection of the curve pair in Figure 6.6 determines the center of the bell-shaped transfer characteristics. As shown in Figure 6.6a, the slopes at the intersection point are dependent on the common-mode charge while the value of ΔV_{in} at the intersection is not. Therefore, the common-mode charge can be programmed to tune the width of the bell-shaped transfer characteristics without affecting the center. On the other hand, the value of ΔV_{in} at the intersection is shifted as the differential charge is changed, but the slopes at the intersection are invariant, as shown in Figure 6.6b. Thus, by programming the differential charge, the center of the transfer function can be tuned without altering the width. Because the template information is stored in a pair of floating-gate transistors as in [13,37], this circuit can be used to implement adaptive learning algorithms with not only an adaptive mean but also an adaptive variance.

The *n*MOS transistors in the VGA are assumed to be in the transition between the above-threshold and the subthreshold regions. The *p*MOS transistors are assumed to be in the above-threshold region. Because the transfer characteristics of the two branches are symmetric, the half circuit technique is used to analyze the VGA gain. By equating the currents flowing through the *p*MOS and *n*MOS transistors, the following expression can be derived:

$$I_{0,\mathrm{p}}\left(\frac{W_{\mathrm{p}}}{L_{\mathrm{p}}}\right)\frac{1}{4U_{\mathrm{T}}^2}\left[\kappa_{\mathrm{p}}(V_{\mathrm{DD}}-V_{\mathrm{fg},21}-V_{\mathrm{T0,p}})\right]^2 = I_{0,\mathrm{n}}\left(\frac{W_{\mathrm{n}}}{L_{\mathrm{n}}}\right)\ln^2\left(1+e^{\frac{\kappa_{\mathrm{n}}}{2U_{\mathrm{T}}}(V_1-V_{\mathrm{T0,n}})}\right), \qquad (6.11)$$

where the variables with the subscript of "p" or "n" are parameters associated with a *p*MOS or a *n*MOS transistor respectively, I_0 is the subthreshold pre-exponential current factor, and V_{T0} is the threshold voltage. At the peak of the bell-shaped transfer curve, $V_{\mathrm{Q,dm}} = 0$, $\mathrm{V}_{\mathrm{fg},21} = \Delta V_{\mathrm{in}}/2 + V_{\mathrm{Q,cm}}$, and $V_1 = V_{\mathrm{out,cm}} + \Delta V_{\mathrm{out}}/2$, where $V_{\mathrm{out,cm}} = (V_1 + V_2)/2$ and $\Delta V_{\mathrm{out}} = V_1 - V_2$. The gain of the VGA can be derived by differentiating (6.11) with

respect to $V_{\mathrm{fg},21}$. Then, the following equation is generated:

$$\begin{aligned}\frac{\Delta V_{\mathrm{out}}}{\Delta V_{\mathrm{in}}} &= \frac{dV_1}{dV_{\mathrm{fg},21}} = -\gamma\left(1 + e^{-\frac{\kappa_{\mathrm{n}}}{2U_{\mathrm{T}}}(V_1 - V_{\mathrm{T0,n}})}\right) \\ &= \frac{-\gamma}{1 - e^{-\frac{\gamma\kappa_{\mathrm{p}}}{2U_{\mathrm{T}}}(V_{\mathrm{DD}} - V_{\mathrm{fg},21} - V_{\mathrm{T0,p}})}} \\ &\approx -\gamma\left(1 + e^{-\frac{\gamma\kappa_{\mathrm{p}}}{2U_{\mathrm{T}}}(V_{\mathrm{DD}} - V_{\mathrm{Q,cm}} - V_{\mathrm{T0,p}})}\right),\end{aligned} \tag{6.12}$$

where $\gamma = \frac{\kappa_{\mathrm{p}}}{\kappa_{\mathrm{n}}}\sqrt{\frac{I_{0,\mathrm{p}} W_{\mathrm{p}} L_{\mathrm{n}}}{I_{0,\mathrm{n}} L_{\mathrm{p}} W_{\mathrm{n}}}}$. Therefore, the gain increases approximately exponentially with the common-mode charge. Hence, an exponential relationship between the extracted standard deviation of the transfer curve and the common mode charge is expected.

If the transistors in the conventional bump circuit are properly sized and the VGA gain is expressed as η,

$$\frac{\Delta V_{\mathrm{out}}}{\Delta V_{\mathrm{in}}} \approx -\gamma\left(1 + e^{-\frac{\gamma\kappa_{\mathrm{p}}}{2U_{\mathrm{T}}}(V_{\mathrm{DD}} - V_{\mathrm{Q,cm}} - V_{\mathrm{T0,p}})}\right) = \eta, \tag{6.13}$$

then the transfer function of the complete bump circuit can be expressed as

$$\begin{aligned}I_{\mathrm{out}} &= \frac{4I_{\mathrm{b}}}{2 + e^{\kappa\eta\Delta V_{\mathrm{in}}/U_{\mathrm{T}}} + e^{-\kappa\eta\Delta V_{\mathrm{in}}/U_{\mathrm{T}}}} \\ &= I_{\mathrm{b}}\mathrm{sech}^2\left(\frac{\kappa\eta\Delta V_{\mathrm{in}}}{2U_{\mathrm{T}}}\right)\end{aligned} \tag{6.14}$$

that is used to approximate a Gaussian function. By adjusting $V_{\mathrm{Q,cm}}$, the magnitude of the VGA gain increases exponentially and hence the width of the bell-shaped transfer curve decreases exponentially.

An prototype chip containing a small array of these floating-gate bump circuits was fabricated in a 0.5 μm CMOS process and has been tested. A micrograph of this chip is shown in Figure 6.7. In Figure 6.10a, the common-mode charge is programmed to several different levels and the transfer curves with different widths are measured. The bell-shaped curves are compared with their correspondent Gaussian fits. In Figure 6.10a, the extracted standard deviation varies 5.87 times and the mean is only shifted by 4.23%. In a semi-logarithmic plot of Figure 6.10b, the extracted standard deviation, σ, is exponentially dependent on the common-mode charge, as predicted by (6.13). From this prototype chip, the measured

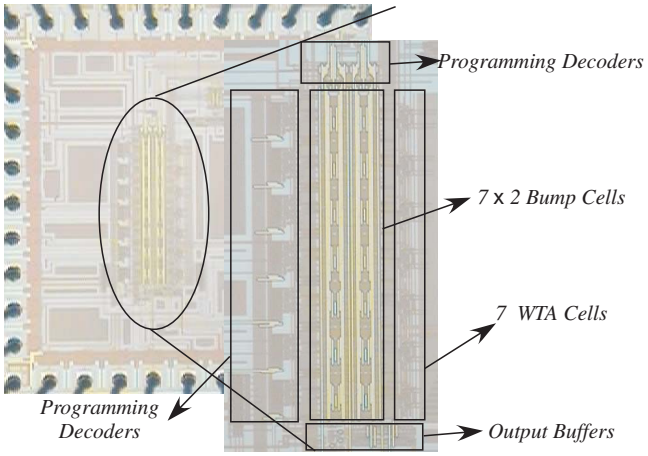


Figure 6.7. The micrograph of a prototype floating-gate bump circuit chip.

minimum achievable extracted standard deviation is 0.199V, which is set by the maximum gain of the VGA. If more than one diode-connected *n*MOS transistor is used as the load in the VGA, then the maximum VGA gain will increase and the minimum achievable standard deviation will be reduced.

6.4 Multivariate Analog RBF Implementation

A multivariate Gaussian function with a diagonal covariance matrix can be approximately realized by cascading these floating-gate bump circuits. To illustrate this, a bivariate Gaussian function can be expressed and be approximated as

$$\begin{aligned} f(\Delta x, \Delta y) &= e^{-\frac{\Delta x^2}{2{\sigma_x}^2}-\frac{\Delta y^2}{2{\sigma_y}^2}} \\ &= e^{-\frac{\Delta x^2}{2{\sigma_x}^2}} \cdot e^{-\frac{\Delta y^2}{2{\sigma_y}^2}} \\ &\approx I_{\text{out}}(\Delta V_{\text{x}}, \Delta V_{\text{y}}) \\ &= I_{\text{b}} \cdot \text{sech}^2\left(\frac{\kappa\eta_x \Delta V_{\text{x}}}{2U_{\text{T}}}\right) \cdot \text{sech}^2\left(\frac{\kappa\eta_y \Delta V_{\text{y}}}{2U_{\text{T}}}\right) \\ &= I_{\text{out,x}} \cdot \text{sech}^2\left(\frac{\kappa\eta_y \Delta V_{\text{y}}}{2U_{\text{T}}}\right), \end{aligned} \tag{6.15}$$

where I_{b} is the input tail current of the first stage, ΔV_{x} and ΔV_{y} are the input signals of the first and second stages, respectively, η_x and η_y are the values of the VGA gain. An example of a bivariate radial basis function implementation is shown in Figure 6.9. The first stage

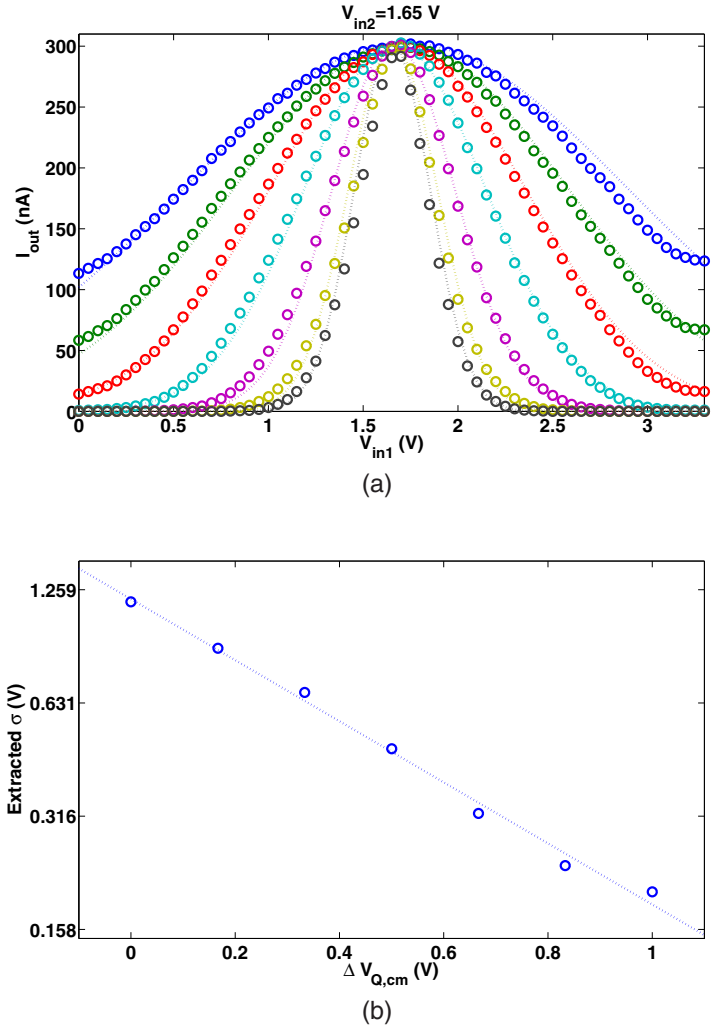


Figure 6.8. Gaussian fits of the transfer curves and the width dependance.
(a) Comparison of the measured 1D bumps (circles) and the corresponding Gaussian fits (dashed lines)
(b) The width versus the common-mode charge on a semi-logarithmic scale

output current is converted to a voltage by a diode-connected transistor, M_{37}. This output voltage is fed to a tail transistor, M_{30}, in the next stage. The output current of the final stage can approximate a multivariate Gaussian function with a diagonal covariance matrix. The maximum value of the output current is set by the first stage tail current, I_b. The feature dimension can be increased by cascading more floating-gate bump circuits but with the cost of the reduction of the classifier bandwidth. The mismatches between the floating-gate bump circuits can be trimmed out by using floating-gate programming techniques. Two measured bivariate "bumps" with different widths are shown in Figure 6.10. By summing

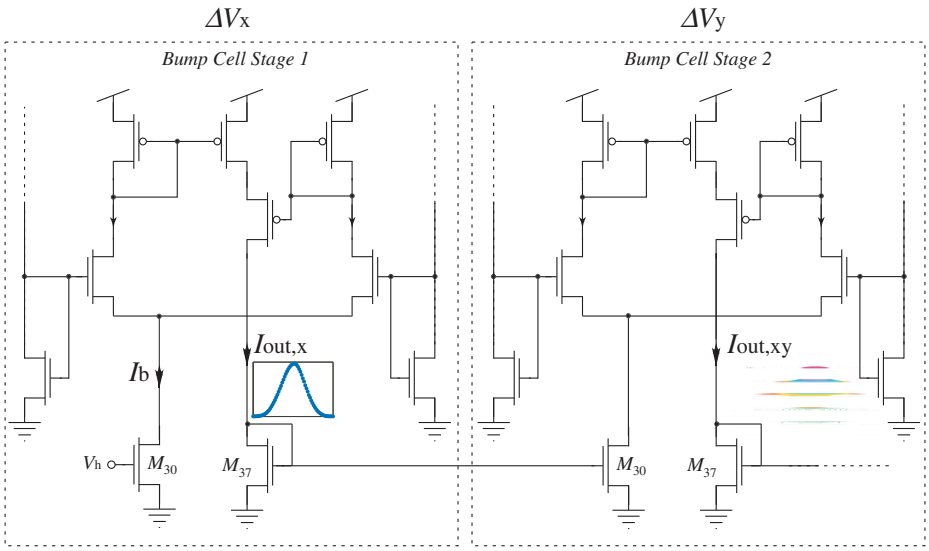


Figure 6.9. A bivariate radial basis function implementation.

up the output currents of an array of these floating-gate bump circuits, Gaussian mixture models can be easily implemented.

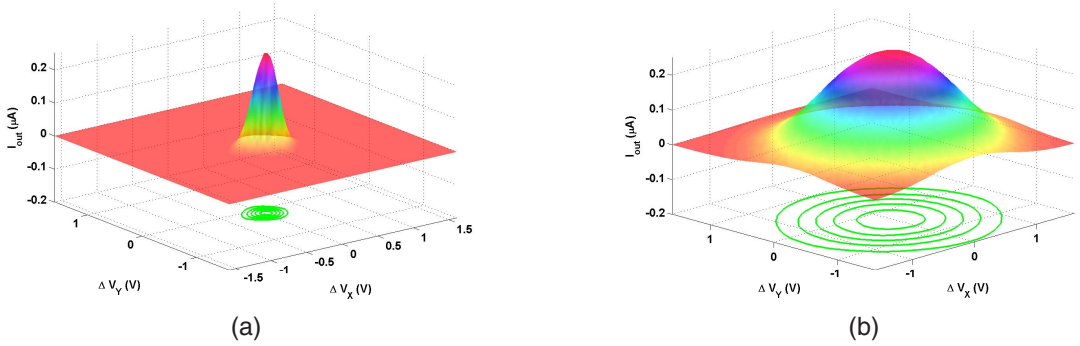


Figure 6.10. Measured results of the bivariate radial basis function implementation.

6.5 Conclusion

Based on previous works, a fully-programmable floating-gate bump circuit is proposed to implement Gaussian functions. The height, the width, and the center of its transfer curve, which are respectively related mathematically to the maximum likelihood, the variance, and the mean of a distribution, are independently programmable. Cascading several

floating-gate bump circuits can implement a multi-variate Gaussian function with a diagonal covariance matrix. A detail description and analysis, as well as the measurement results, are given in this chapter. Based on this floating-gate bump circuit, a multi-variate programmable analog RBF-based classifier presented in the next chapter shows great power efficiency compared with digital signal processors.

CHAPTER 7

A PROGRAMMABLE ANALOG RBF-BASED CLASSIFIER

A programmable multi-dimensional analog RBF-based classifier is presented in this chapter. The probability distribution of each feature in classifier templates is modeled by a Gaussian function. The bell-shaped transfer characteristics of a previously introduced programmable floating-gate bump circuit approximate a Gaussian function. The maximum likelihood, the mean, and the variance of the distribution are stored in floating-gate transistors and are independently programmable. By cascading the floating-gate bump circuits, the overall transfer characteristics approximate a multivariate Gaussian function with a diagonal covariance matrix. An array of these circuits constitute a compact multi-dimensional RBF-based classifier that can easily implement a Gaussian mixture model. When followed by a winner-take-all circuit, the RBF-based classifier forms an analog vector quantizer. We use receiver operating characteristic curves and equal error rate to evaluate the performance of our analog RBF-based classifier as well as a resultant analog vector quantizer. Automatic gender identification experiment was conducted on a 16×16 analog vector quantizer chip to demonstrate one possible audio application of this work. We show that the analog classifier performance is comparable to that of digital counterparts. The power efficiency of this analog approach is at least two orders of magnitude better than that of digital microprocessors at the same task.

7.1 Information Refinement in The Analog Domain

The fabrication and packaging technologies have enabled an unprecedented number of components to be packed into a small volume and the resultant power density can be higher than ever. The power dissipation has become the bottleneck factor in determining both the functionality and the feasibility of a sensory microsystem. Conventionally, all the classification, decision-making, or, in more general terms, information-refinement tasks are

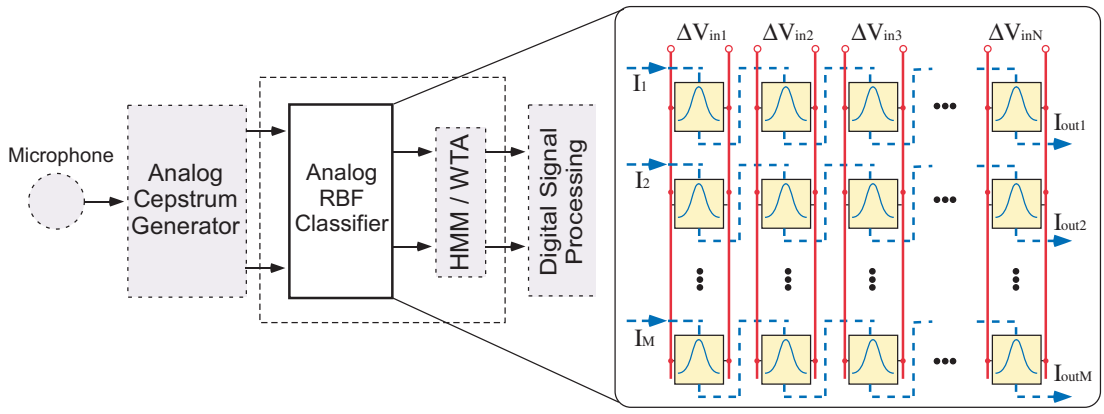


Figure 7.1. The block diagram of an analog system for speech recognition.

performed in a digital processor. If the information-refinement tasks can be performed efficiently in the analog domain at the interface, the information bandwidth in the subsequent stages can be reduced and the specifications for analog-to-digital-converters, which are usually power-hungry, can also be relaxed. In this manner, the power consumption of the microsystems can be further decreased.

An analog speech recognizer proposed in [39] is an example of performing the information-refinement tasks in the analog domain. Its block diagram is shown in Figure 7.1. The recognizer frontend includes a band-pass-filter-bank-based analog Cepstrum generator, an analog RBF-based classifier, and a continuous-time hidden Markov model (HMM) block that is built from programmable analog waveguide stages. The input to the HMM stage could represent the RBF response directly or it could pass through a logarithmic element first. The HMM stage can also be replaced by a winner-take-all (WTA) block in some applications. By performing analog signal processing at the front end, the required specifications for analog-to-digital converters can be relaxed in terms of speed, accuracy, or both and the computational load of the subsequent digital processor can be reduced. As a result, the entire system can be more power efficient. The analog RBF-based classifier is a critical building block in this analog system for speech recognition.

7.2 Analog Classifiers

In this chapter, a highly compact and power-efficient, programmable analog RBF-based classifier is demonstrated. As shown in Figure 7.1, the distribution of each feature in the templates is modeled as a Gaussian function and is implemented by a floating-gate bump circuit that has been detailed in the previous chapter. The maximum likelihood, the variance, and the mean of the distribution are respectively associated mathematically with the height, the width, and the center of the floating-gate bump circuit transfer curve. These three parameters can be independently programmed and, hence, the classifiers can fit into different scenarios with the full use of the available statistical information up to the second moment.

When followed by a WTA stage, the RBF-based classifier forms a multi-dimensional analog vector quantizer. A vector quantizer compares the distances or the similarities between an input vector and the stored templates. It classifies the input data as the most representative template. Vector quantization is a typical pattern recognition and data compression technique. Crucial issues of the vector quantizer implementation concern the storage efficiency and the computational cost of searching the best-matching template. In the past decade, efficient digital [40, 41] and analog [35, 36, 42] vector quantizers have been developed. In general, analog vector quantizers have been shown to be more power efficient than digital ones. However, in a previous design [42], the computational efficiency is partially because of the fact that only the mean absolute distances between the input vector and the templates are compared instead of considering the possible feature distributions. To have a better approximation to the Gaussian distribution, many variations of analog RBF circuits have been designed [30–35]. These circuits are usually complicated and require extra circuits to store or to periodically refresh the template data. In [13, 36, 37], floating-gate transistors are used to implement analog classification systems. Because the template data are stored as the charges on floating-gate transistors, the resulting systems are very compact. However, these systems cannot be applied to the non-uniform variance scenarios.

The block diagram of a proposed analog vector quantizer is shown in Figure 7.2. Because the analog RBF-based vector quantizer is composed of an array of fully-programmable floating-gate bump circuits, the stored template information can be closer to the real distributions and the resultant vector quantizer can be applied to non-uniform, as well as uniform, variance scenarios. A current-mode WTA circuit compares output currents of the analog RBF-based classifier and indicates the winning template.

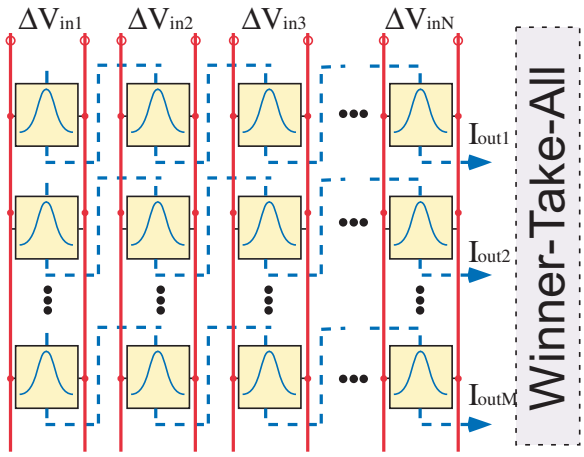


Figure 7.2. The block diagram of an analog RBF-based vector quantizer.

7.3 The Architecture of The Analog Vector Quantizer

The template information of the analog RBF-based classifier needs to be programmed into the floating-gate bump circuits. To do so, all the floating-gate transistors in the vector quantizer should to be arranged as those in a floating-gate programming framework as shown in Figure 2.9. Therefore, several multiplexers are inserted into the floating-gate bump circuit and its bias generation block, as shown in Figure 7.3. The "1" on the multiplexer indicates the connection in the programming mode and the "0" indicates the connection in the operating mode. Tunneling junction capacitors are not shown for simplicity. Most of the multiplexers are in the bias generation and the inverse generation blocks. Only two multiplexers are added in the VGA.

In Figure 7.2, the same input voltage vector is compared with all the stored templates.

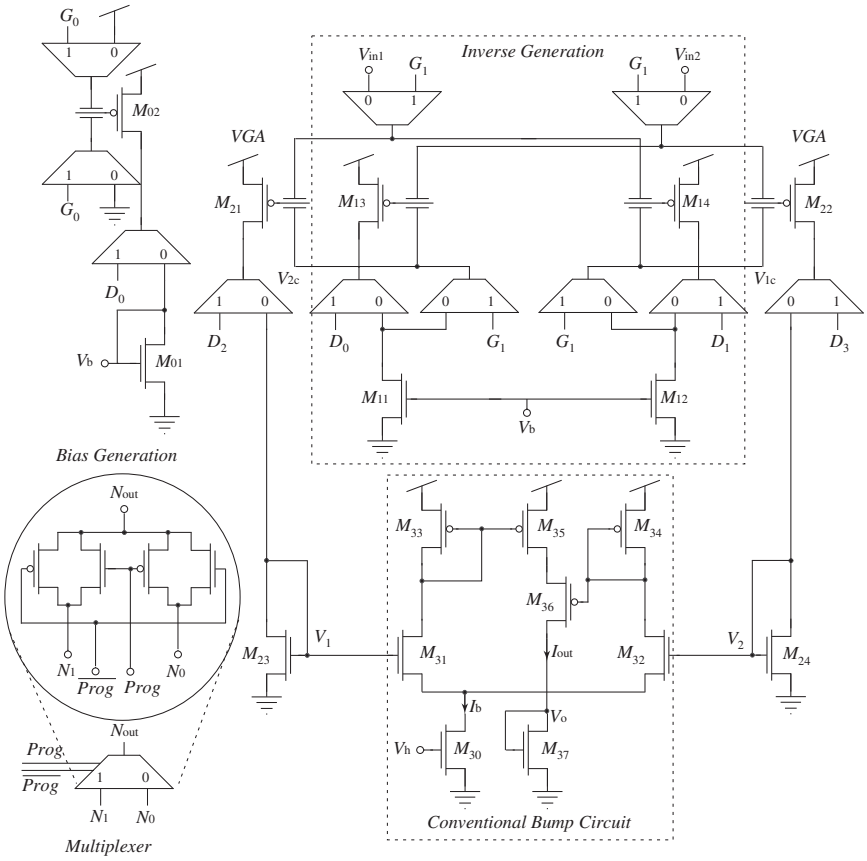


Figure 7.3. Complete schematics of the floating-gate bump circuit

Therefore, the inverse generation block can be shared by the bump circuits in the same column and can be moved to the top of the array. The number of the inverse generation blocks is equal to the dimension of the feature space and is independent of the number of the templates. Only one bias generation block is needed for the entire classifier and it can also be placed at the top of the array. Therefore, most of the multiplexers for floating-gate programming are at the peripheries of the classifier. The element of the resultant bump cell array is composed of a conventional bump circuit and a VGA with two programming multiplexers.

The tail current of a bump cell in the first column sets the maximum likelihood of the corresponding template. To program this current, a *"FG-pFET & Mirror"* block is placed in front of the first bump cell. The schematic of this block is shown in Figure 7.4a. For the analog vector quantizer implementation, the final output currents of the RBF-based

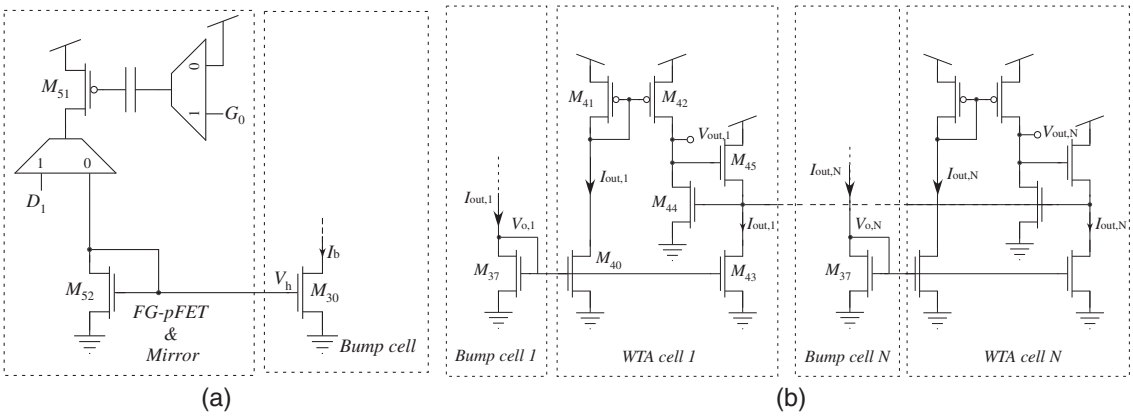


Figure 7.4. The schematics of the "FG-pFET & Mirror" block and the winner-take-all circuit.
(a) "FG-pFET & Mirror" block
(b) Current mode winner-take-all circuit

classifier are duplicated and are fed into a simple current mode WTA circuit. The schematic of the WTA circuit is shown in Figure 7.4b. Only the output voltage of the winning cell will be high to indicate the best-matching template. The final architecture of the proposed analog vector quantizer is shown in Figure 7.5. Together with the gate-line and drain-line decoders, most of the programming overhead circuitry is at the peripheries of the floating-gate bump cell array; therefore the system can be easily scaled up and still maintains high compactness. The compactness and the ease of scaling up are important issues in the implementation of an analog speech recognizer that requires more than a thousand bump cells.

7.4 A Prototype Analog Vector Quantizer

A prototype chip containing a 7×2 programmable analog vector quantizer was fabricated in a 0.5 μm CMOS process and tested. A micrograph of the chip is shown in Figure 6.7. Some important parameters and measured results are listed in Table 7.1.

Four templates are used to demonstrate the reconfigurability of the analog classifiers. Floating-gate transistors in unused templates are tunneled off. Four bell-shaped output currents that approximate the bivariate Gaussian likelihood functions of four templates are overlaid in Figure 7.6. The thick solid lines at the bottom indicate the boundaries

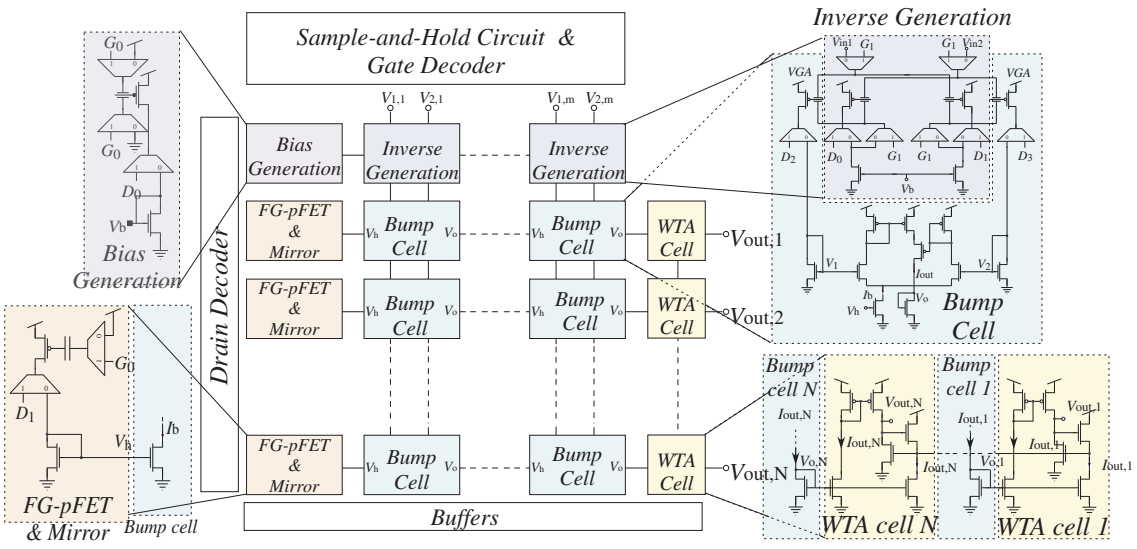


Figure 7.5. The architecture of an analog vector quantizer.

Table 7.1. Analog Vector Quantizer Parameters and Performance

Size of VQ	7(templates)×2(components)
Area/Bump Cell	$42 \times 82\,\mu\text{m}^2$
Area/WTA Cell	$20 \times 35\,\mu\text{m}^2$
Power Supply Rail	$V_{\text{DD}} = 3.3V$
Power Consumption/Bump Cell	$90\,\mu\text{W} \sim 160\,\mu\text{W}$
Response Time	$20\mu \sim 40\mu\text{sec}$
Floating-gate Programming Accuracy	99.5%
Retention Time	10 years @ 25°C

determined by the WTA outputs. V_{X} and V_{Y} are the values of V_{in1} in Figure 7.3 applied to the first and the second floating-gate bump circuit, respectively. The values of V_{in2} in both stages are fixed at $V_{\text{DD}}/2$.

7.4.1 Power consumption

To evaluate the power consumption, several "bumps" are programmed with an identical width and unused bump cells are deactivated by tunneling their floating-gate transistors off. The power consumption is averaged over the entire two-dimensional input space. The slope

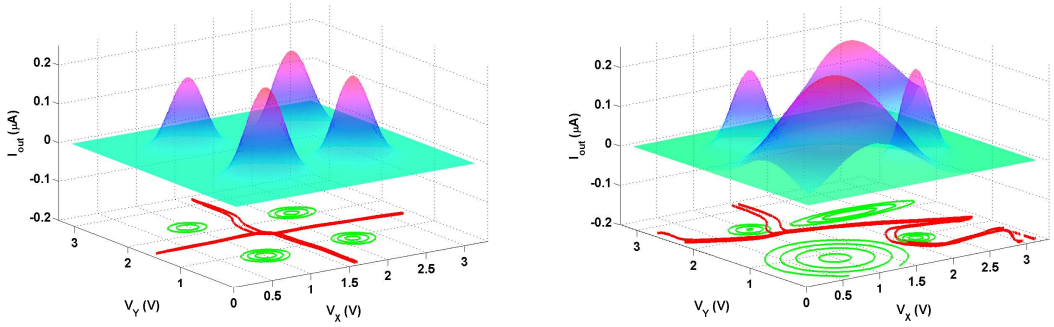


Figure 7.6. Configurable classification results.

of the curve in Figure 7.7a indicates the average power consumption per bump cell for a specific value of the width. The relation between the power consumption and the extracted standard deviation is plotted in Figure 7.7b: The wider the "bump" is, the more the power consumption.

The VGA is the major source of power consumption. The gain is tunable when the *n*MOS transistors operate in the transition between the above-threshold and the subthreshold regions. The width tunability can also result from the nonlinearity of the *p*MOS transistors when they are in transition between the saturation and the ohmic regions. From simulation, to reduce the power consumption, we can make *n*MOS transistors in the VGA long to reduce the above-threshold currents and raise the source voltages of M_{23} and M_{24} to reduce the headroom.

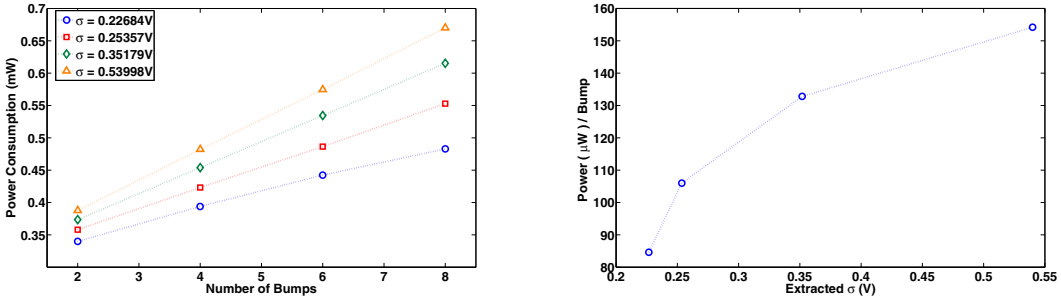
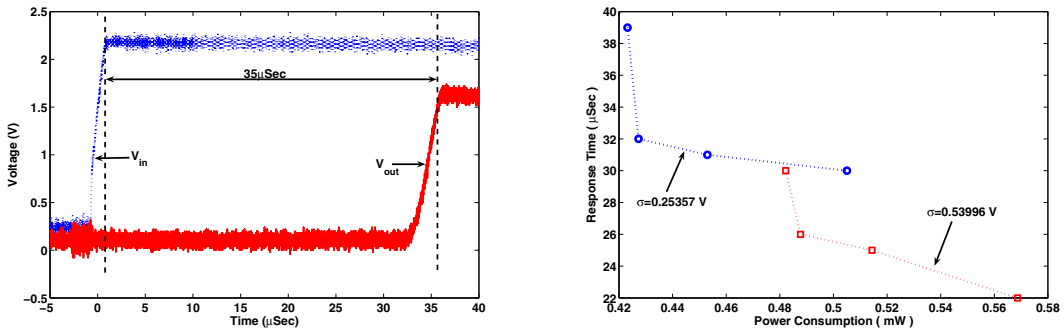


Figure 7.7. The relation between the power consumption and the extracted standard deviation.
(a) The average power consumption versus the number of activated floating-gate bump cells
(b) The average power consumption versus the extracted standard deviation

7.4.2 Bandwidth

The bandwidth of the floating-gate bump circuit is expected to be around mega-Hz range and cannot be directly measured from the prototype chip. This is because the RBF output current is in the nano-amp range and the bandwidth of the current preamplifier for measurement is approximately 1K Hz at that current level. Only the response time from the input to the WTA outputs can be measured. Measured transient response is shown in Figure 7.8a. One of the speed bottlenecks of the system comes from the inverse generation block. For a given width, the speed and the power depend on the amount of charge on M_{13} and M_{14}. With more electrons on the floating gates, the circuit can achieve higher speed but consumes more power. The speed-power relation is shown in Figure 7.8b. The steep region of the curve implies that the inverse generation block dominates the circuit bandwidth. In this region, we can increase the speed by consuming more power in the inverse generation block. The flat region in Figure 7.8b indicates the VGA dominant region. In this region, burning more power in the inverse generation block does not improve the system bandwidth. Thus, given a value of the standard deviation, the tradeoff between the speed and the power consumption in the inverse generation block can be optimized by programming the charges on M_{13} and M_{14} to have the system operate at the knee of the curve.



(a) (b)

Figure 7.8. The transient response and the speed-power tradeoff.
(a) The transient response of the vector quantizer
(b) The response time versus power consumption

7.4.3 Performance

The performance of the analog RBF computation is evaluated. Since the computation method and errors are different from those of traditional digital approaches, generic comparisons of effective bit-accuracy do not make sense. Rather, we choose to evaluate the impact of using the analog RBF on system performance. To this end, receiver operating characteristic (ROC) curves are adopted to characterize the classifier performance. The ROC graphs indicate the whole range of the operating characteristics and provide a richer measure of classification performance than scalar measures such as accuracy, error rate, or error cost. Because ROC graphs decouple classifier performance from class skew and error costs, they have advantages over other evaluation measures.

Two separate two-dimensional "bumps" are programmed to have the same variance with a separation of 1.2 V, as shown in Figure 7.9. The corresponding Gaussian fits are used as the actual probability density functions of two classes. Comparing these two probability density functions using different threshold values generates an ROC curve of these two Gaussian distributed classes. We use this ROC curve as the evaluation reference. With the knowledge of the class distributions, comparing the output currents using different threshold values generates an ROC curve for the analog RBF circuits. Comparing each of the two WTA output voltages with different threshold values generates two ROC curves that characterize the classification results from the vector quantizer. The equal error rate (EER), which is the intersection of the ROC curve and the -45° line, is the usual operating point of the classifiers. The ROC curves and the EER comparisons are shown in Figure 7.10. In Figure 7.11, both the ROC areas and the EER are plotted to investigate the effect of the width on the classifier performance. At the EER point, the performance of the analog RBF classifier, which uses floating-gate bump circuits to approximate Gaussian likelihood functions, is undistinguishable from that of an ideal RBF-based classifier. Despite the finite gain of the WTA circuit, the performance of the analog vector quantizer is still comparable to an ideal maximum likelihood (ML) classifier. By optimizing the precision and the speed

of the WTA circuit, the classifier performance can be improved.

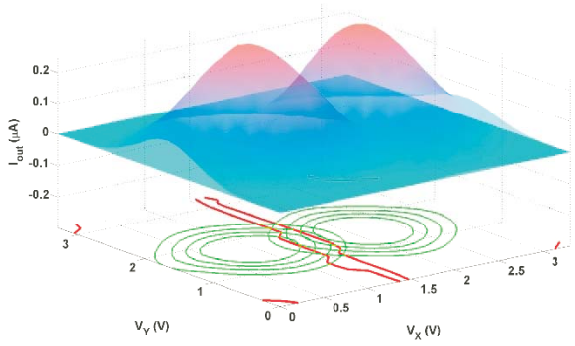


Figure 7.9. The distributions of two "bumps" used to evaluate the classifier performance.

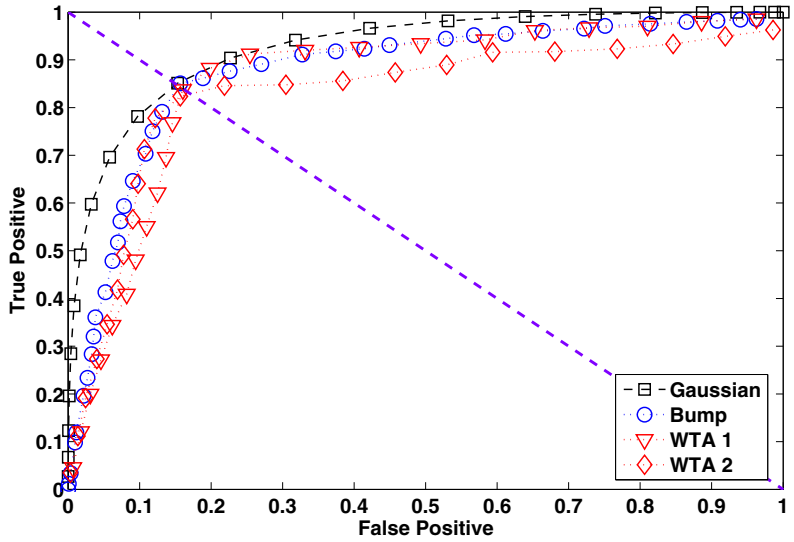


Figure 7.10. The ROC curves of the Gaussian fits (squares), the output currents of the two-dimensional bumps (circles) and the WTA output voltages (triangles and diamonds) with the extracted standard deviation of 0.55 V.

7.4.4 Power efficiency

To compare the efficiency of the analog system with DSP hardware, the metric of millions of multiply accumulates per second per milli-watt (MMAC/s/mW) is estimated. Only the performance of a single bump cell is considered because when the system is scaled up, the bump cells dominates the system efficiency.

Each Gaussian function is estimated as 10 MACs and can be evaluated by a bump cell

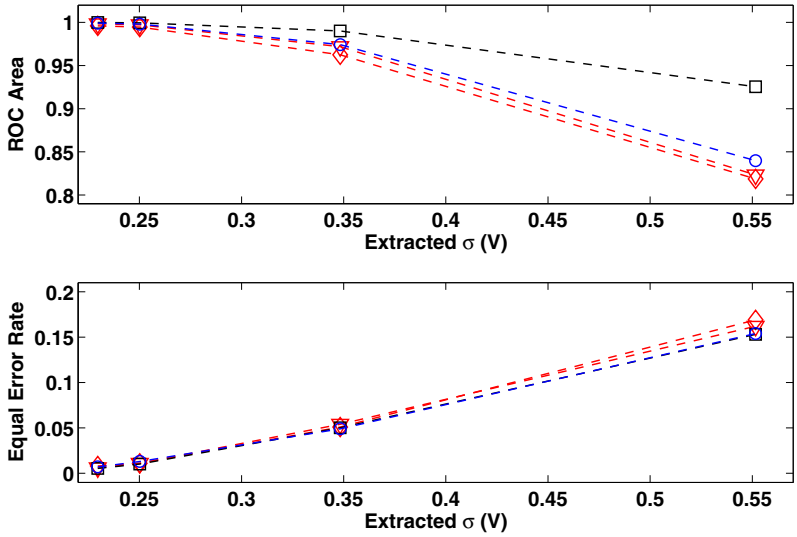


Figure 7.11. The effects of the bump widths on the ROC area and the EER performance.

in less than 10μ sec (which is still an overestimate) with the power consumption of $120\,\mu$W. This is equivalent to 8.3 MMAC/s/mW. The performance of commercial low-power DSP microprocessors ranges from 1 MMAC/s/mW to 10 MMAC/s/mW and a special designed high performance DSP microprocessor in [43] is better than 50 MMAC/s/mW. If this comparison is expanded to include the WTA function, the efficiency of our analog classification system will improve even more relative to the digital system.

7.5 A 16×16 Analog Vector Quantizer

Based on the measurement results from the prototype chip, a 16×16 analog vector quantizer has been designed and fabricated in a $0.5\,\mu$m CMOS process occupying silicon area less than $1.5 \times 1.5mm^2$. A micrograph of the chip is shown in Figure 7.12. In this chip, two diode-connected nMOS transistors are used as the load in the VGA to reduce the minimum achievable width of the transfer curve. Cascading more diode-connected transistors also reduces power consumption.

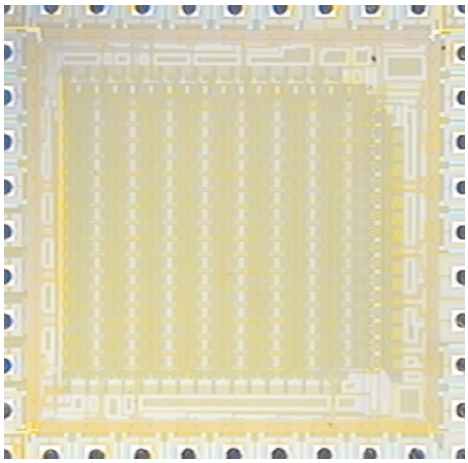


Figure 7.12. The micrograph of the 16 × 16 analog vector quantizer.

7.5.1 Programming accuracy

Although the floating-gate bump circuit can be applied to both single and differential input scenarios, in the following measurements, differential input voltages are used as the input signals to double the input dynamic range. The common-mode charge of one floating-gate bump circuit is programmed to have different values and measured transfer curves and corresponding Gaussian fits are shown in Figure 7.14. The minimum achievable standard deviation of this version of the floating-gate bump circuit is 40 mV and the relation between the common-mode charge related voltage and the extracted standard deviation is shown in Figure 7.14.

After the characterization process, 16 different floating-gate bump circuits in one template can be precisely programmed to have linearly variant widths, as shown in Figure 7.15. The offsets of these 16 bump circuits are within 26mV, as shown in Figure 7.18a. Measured standard deviations are compared with the targets in Figure 7.15b. The programmed standard deviation errors are less than 5%, as shown in Figure 7.18b.

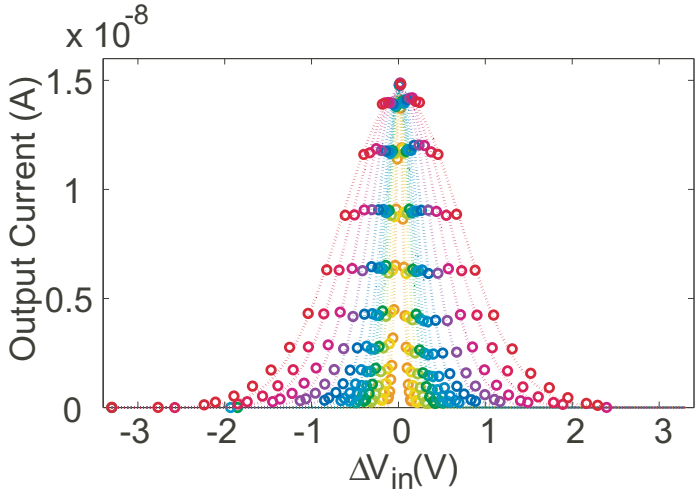


Figure 7.13. The transfer curves of a floating-gate bump circuit in the analog vector quantizer.

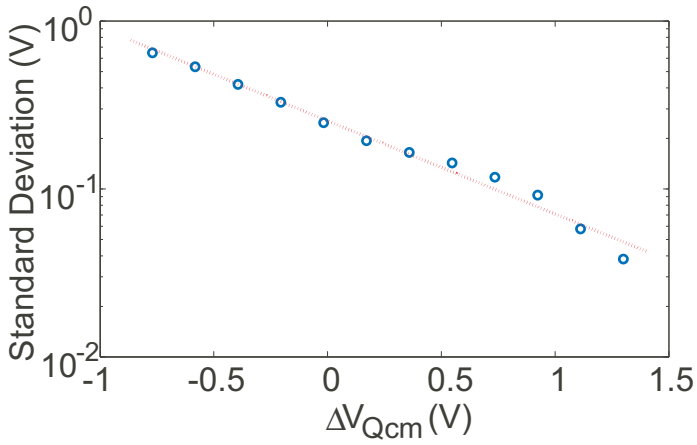


Figure 7.14. The common-mode charge and the extracted standard deviation in the analog vector quantizer.

7.5.2 Performance

Any two of the 16 bump circuits in the same template can be swept in a two-dimensional space while others remain constant to visualize the resultant bivariate distribution in a three-dimensional plot. Two examples of these plots measured from the floating-gate bump circuits that are programmed as in Figure 7.15a are shown in Figure 7.17.

The centers and the widths of all 16 templates can be precisely programmed in a 16-dimensional feature space. The maximum likelihoods of 16 templates can also be adjusted. In Figure 7.19, all 16 templates are programmed to have same width and height. Two of the

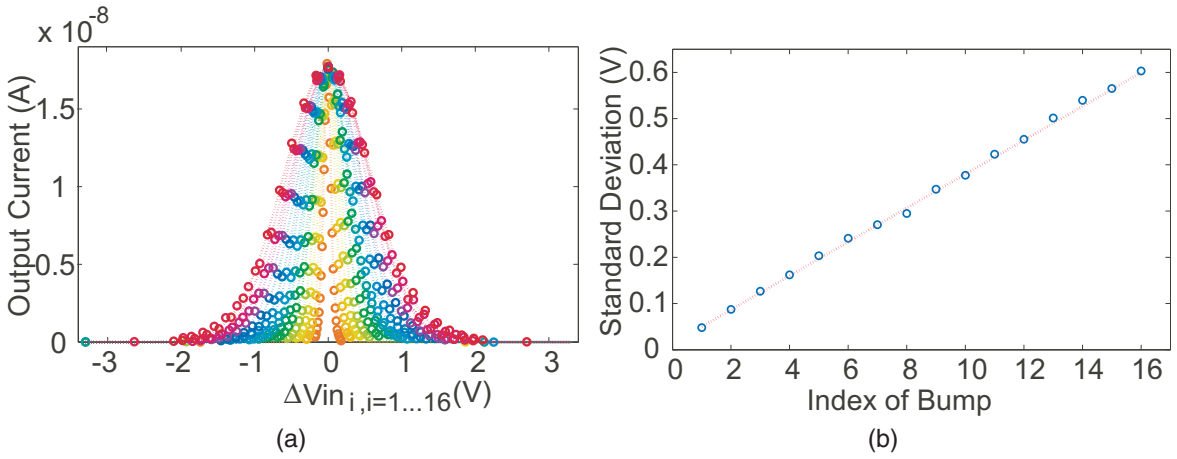


Figure 7.15. The transfer curves of 16 different floating-gate bump circuits in the same template.
(a) The measured transfer curves
(b) The extracted standard deviations

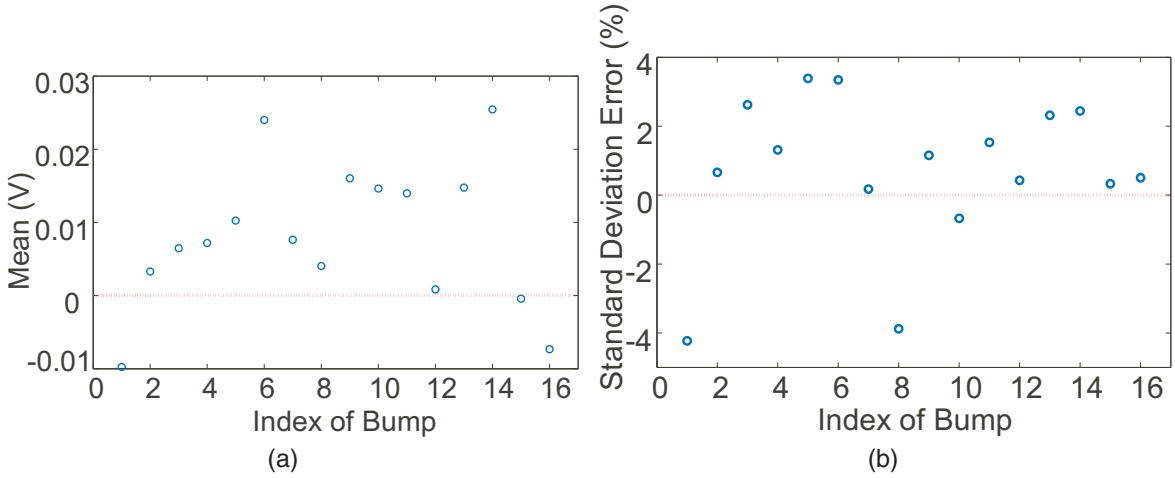


Figure 7.16. The programming accuracy of the 16 different floating-gate bump circuits in the same template.
(a) The offsets of the center
(b) The errors of the extracted standard deviations

16 features are programmed so that 16 templates are evenly spaced in a two-dimensional space. The output currents and the WTA output voltages are overlaid in a single plot. The thick lines at the bottom plane indicate the boundaries determined by the WTA circuit.

To characterize the classifier performance, two templates are programmed to have identical variance of 0.5V with separation of 1V. The ROC curves are shown in Figure 7.20 for comparison. The ROC areas under four curves are 0.921, 0.869, 0.898, and 0.876, respectively. The EERs of these four curves are 0.160, 0.160, 0.159, and 0.159 respectively.

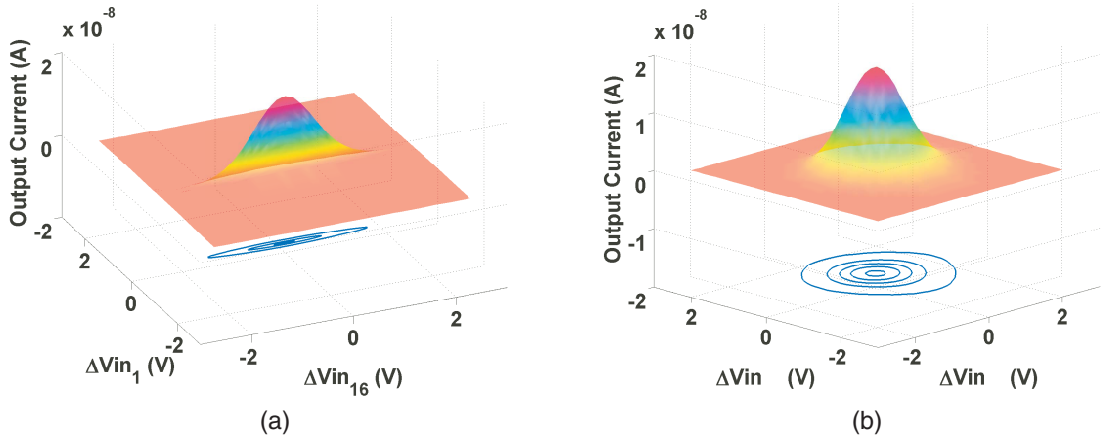


Figure 7.17. Two-dimensional "bumps".
(a) The first and the 16th bumps programmed in Figure 7.15aare swept
(b) The 15th and the 16th bumps programmed in Figure 7.15a are swept

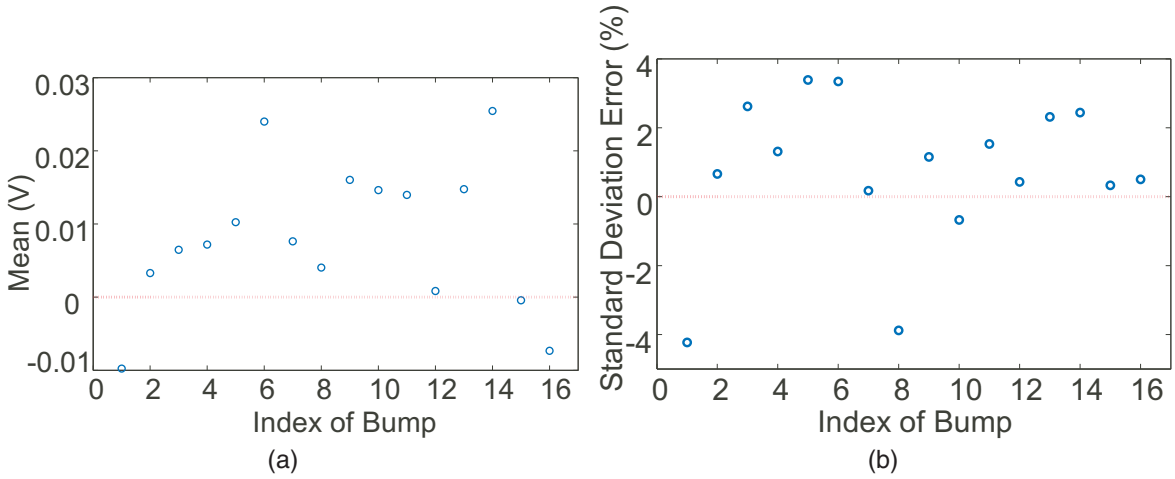


Figure 7.18. The programming accuracy of the 16 different floating-gate bump circuits in the same template.
(a) The offsets of the center
(b) The errors of the extracted standard deviations

7.5.3 Automated gender identification

An automatic gender identification (AGI) task is performed on this 16×16 analog vector quantizer chip to demonstrate one possible audio application of this work. AGI systems have been used in many automatic speech or speaker recognition systems to enhance their

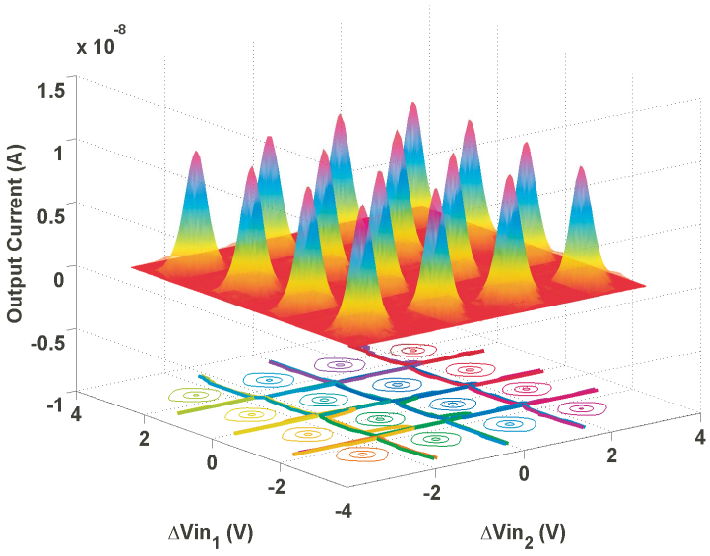


Figure 7.19. The transfer curves of a floating-gate bump circuit in the analog vector quantizer.

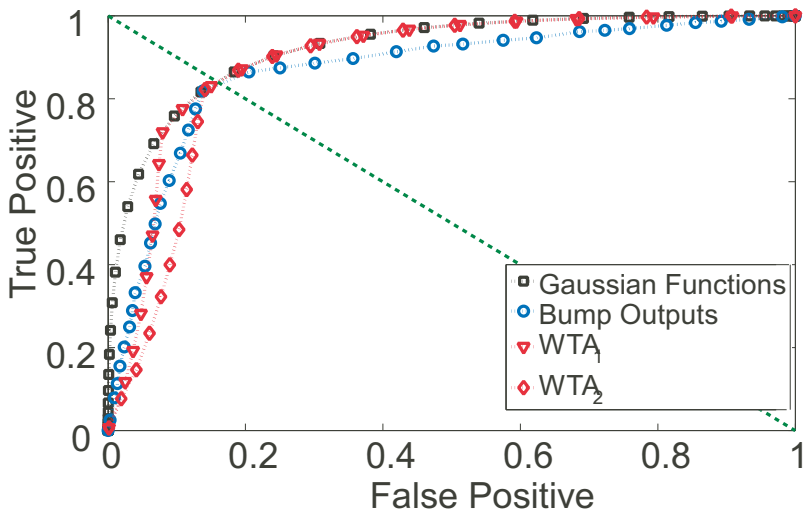


Figure 7.20. The ROC curves measured from the 16 × 16 analog vector quantizer.

performance.

With the available number of templates and feature dimensions, the analog vector quantizer chip implements a simple AGI classifier. Eight 14-variate Gaussian components are used to characterize one specific gender, and a winner-take-all voting scheme makes the final decision. The experiment is conducted on the Aurora-2 database [44], which provides independent training and testing sets and is a standardized database for speech recognition

research. Four hundred utterances from the training set recorded in clean conditions are used to train the models by means of the maximum likelihood criterion. The speech data is windowed to 100-ms frames and parameterized into 14 order MFCCs, consisting of 13 cepstral coefficients along with a logarithmic energy value. Although these features are prepared from a computer in this demonstration, they can be provided from an analog Cepstrum generator, as proposed in [39]. Therefore, a highly power-efficient complete analog audio recognizer frontend is feasible.

One thousand utterances from the testing set are used to evaluate the performance. The confusion matrix is presented in Table 7.2. The accuracy of the ideal model on the testing set is 73.7% and the accuracy obtained from the analog vector quantizer is 69.8%. One reason of the performance degradation is the limited programmable range of mean and variance. If the charge on the floating-gate transistor in the programming framework is too much, when the control gate are connected to V_{DD}, the channel current is no longer negligible. In this case, the current isolation rule of the programming framework will be violated. The dotted region in Figure 7.21 is the prohibitive region where and values of mean and variance cannot be programmed. Therefore, all the input data need to be scaled down dramatically so that the mean-variance points can be located inside the allowable region. In Figure 7.21, the squares represent the male templates and the circles represent the female templates; different colors stands for different features. The version of this analog classifier chip suffers from resolution issues. This limitation can be overcome by using a pFET switch to turn off the channel current while a floating-gate transistor is not selected for programming. In that case, the performance of the analog classifier is expected to be improved and be comparable to that achieved by digital processors.

7.5.4 Power efficiency

The power consumption of a single bump cell is measured and is shown in Figure 7.22. In this chip, the speed of a single floating-gate bump cell stage can be estimated indirectly

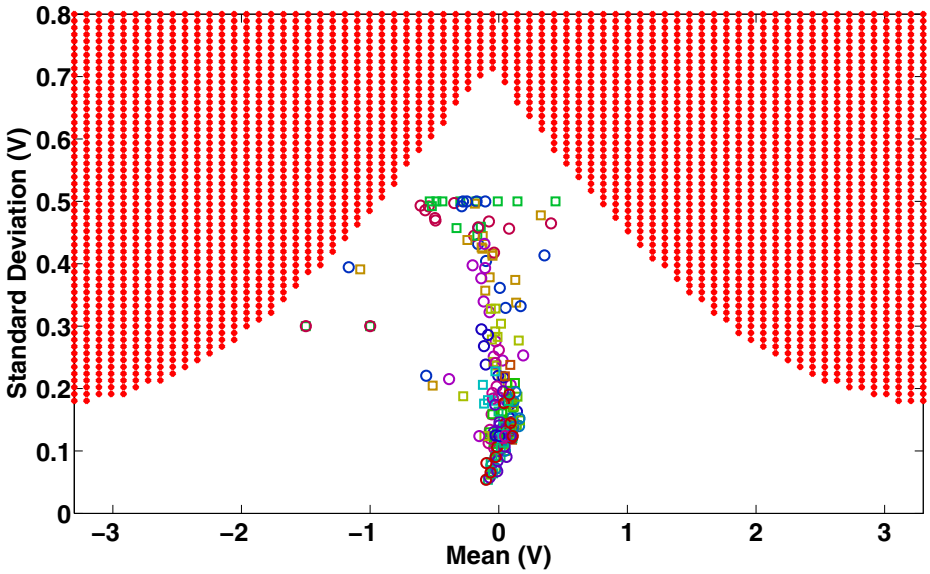


Figure 7.21. The effective region for programmable means and variances.

Table 7.2. AGI Results

	Digital Gaussian Classifier		Analog Vector Quantizer	
	Counts as Male	Counts as Female	Counts as Male	Counts as Female
Present Male	389	111	374	126
Present Female	152	348	176	324

by applying input step from different bump cell stages and differentiating the measured response time. When the maximum output current is programmed to be 100 nA, the response time of a single bump cell is estimated as 0.65μs. If each Gaussian function is estimated as 10 MACs, it can be evaluated by a bump cell in 0.65μs with the power consumption of approximately 30μW. This is equivalent to 513 MMAC/s/mW. The performance of commercial low-power DSP microprocessors ranges from 1 MMAC/s/mW to 10 MMAC/s/mW. If the comparison is expanded to include the WTA function and if the WTA circuit is also optimized, the efficiency of this analog approach can be at least two to three orders of magnitude better than digital microprocessors at the same task. Moreover, this power analysis has not included the power reduction of analog-to-digital converters, which is a major

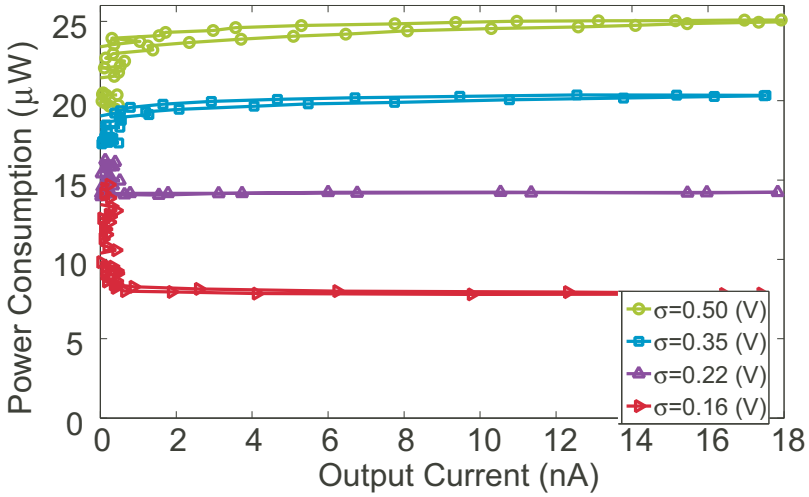


Figure 7.22. The power consumption of a bump cell with different widths.

factor.

7.6 Conclusion

In this chapter, a novel compact programmable analog RBF-based soft classifier and a resultant analog vector quantizer are demonstrated. The performance of the analog classifiers evaluated by ROC graphs and the equal error rate are comparable to digital systems. An automated gender identification experiment is conducted on a 16×16 analog vector quantizer to demonstrate one possible application of this work. The measured power efficiency is at least two orders of magnitude better than commercial digital signal processors.

CHAPTER 8

ADAPTIVE BUMP CIRCUIT FOR CLUSTERING

Floating-gate transistors are used as analog memories to store template information in the analog RBF-based classifier presented in the previous chapter. The classifier needs to be programmed before performing classification tasks. In this chapter, an adaptive bump circuit of which the transfer curve can adapt to input signals is proposed. The template information stored on floating gates is modified using Fowler-Nordheim tunneling and channel hot electron injection according to input signals. An adaptive vector quantizer made of an array of these adaptive bump circuits and a winner-take-all circuit can perform unsupervised learning and clustering tasks.

8.1 Dynamics of Two Source-Coupled pFET Synapses

A pFET floating-gate transistor shown in Figure 8.1a has been named as a pFET synapse [13, 39] because it can locally perform learning rules to bidirectionally modify the weight stored on the floating gate using Fowler-Nordheim tunneling and channel hot electron injection mechanisms. If the pFET synapse transistor is in the subthreshold region and the Early effect is negligible, the source current can be expressed in terms of the floating-gate voltage variation, ΔV_{fg}, around the equilibrium current, I_{s0}, as

$$I_{\mathrm{s}} = I_{\mathrm{s0}} e^{\frac{-\kappa \Delta V_{\mathrm{fg}} + \Delta V_{\mathrm{s}}}{U_{\mathrm{T}}}}. \tag{8.1}$$

The dynamics of the tunneling current [13] can be modeled as

$$I_{\mathrm{tun}} = I_{\mathrm{tun0}} e^{(\Delta V_{\mathrm{tun}} - \Delta V_{\mathrm{fg}})/V_{\mathrm{x}}}, \tag{8.2}$$

where I_{tun0} is the quiescent tunneling current and V_{x} is a constant related to the quiescent tunneling and floating-gate voltages. The typical value of V_{x} is around 1 V. The hot electron injection current can be modeled as

$$I_{\mathrm{inj}} = I_{\mathrm{inj0}} \left(\frac{I_{\mathrm{s}}}{I_{\mathrm{s0}}} \right)^{\alpha} e^{-\Delta V_{\mathrm{d}}/V_{\mathrm{inj}}}, \tag{8.3}$$

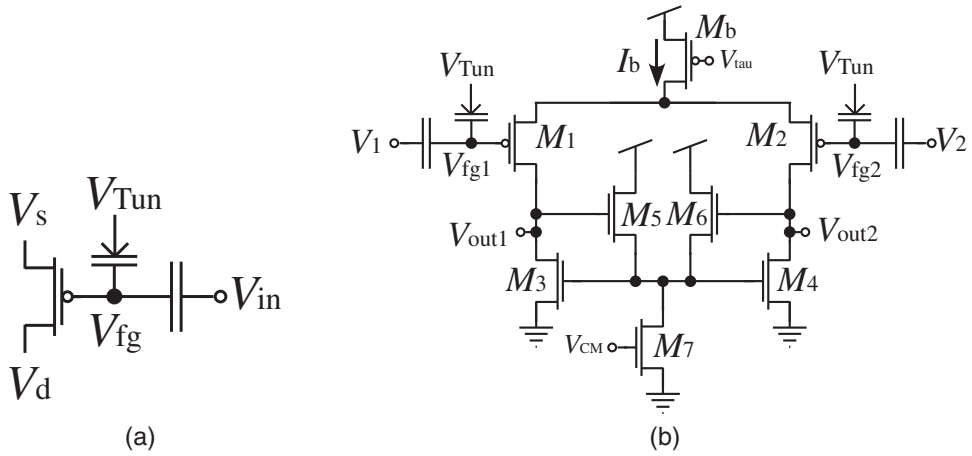


Figure 8.1. The schematics of a pFET synapse and two source-coupled pFET synapses.
(a) A pFET synapse
(b) Two source-coupled synapses

where I_{inj0} is the quiescent injection current, $\alpha = 1 - (U_T/V_{inj})$, and V_{inj} is a device and bias dependent parameter. The typical value of α is 0.9 and the typical value of V_{inj} ranges from 0.2 V to 0.6 V.

A circuit including two source-coupled pFET synapses is used to adapt the template information of a proposed adaptive bump circuit. Two pFET synapses and a pFET tail transistor form a differential pair, as shown in Figure 8.1b, with transistors M_3 and M_4 as the loads. The common-mode feedback is composed of nFET transistors M_{5-7} that keep the common-mode current through M_3 and M_4 half of I_b. Because the circuit is symmetric, in the equilibrium point, the source currents through M_1 and M_2 should be the same. Besides, the tunneling and injection currents should be equivalent. The half circuit technique can be used to analyze this fully differential amplifier. A small perturbation at a floating-gate voltage gives rise to a large variation at the output with a gain of $g_{m1}r_{o3} \approx 1000$. Therefore, the relation between the floating-gate and the drain voltages of a pFET synapse can be given as

$$\Delta V_d = g_{m1}r_{o3}\Delta V_{fg} = A_v \Delta V_{fg}. \tag{8.4}$$

The equilibrium source currents of M_1 and M_2 are defined as I_{s0}. The equilibrium

tunneling current is defined as I_{tun0} and the equilibrium injection current is defined as I_{inj0}. If the source and the tunneling voltages in Figure 8.1b are fixed, the weight of the synapse can be defined as a normalized source current and can be expressed as

$$W = \frac{I_{\text{s}}}{I_{\text{s0}}} = e^{\frac{-\kappa \Delta V_{\text{fg}}}{U_{\text{T}}}}. \tag{8.5}$$

The tunneling and injection currents can also be expressed in terms of the weight, W, as

$$I_{\text{tun}} = I_{\text{tun0}} e^{-\Delta V_{\text{fg}}/V_{\text{x}}} = I_{\text{tun0}} W^{\frac{U_{\text{T}}}{\kappa V_{\text{x}}}}, \tag{8.6}$$

$$I_{\text{inj}} = I_{\text{inj0}} W^{\alpha} e^{A_{\text{v}} \Delta V_{\text{fg}}/V_{\text{inj}}} = I_{\text{inj0}} W^{\alpha - \frac{A_{\text{v}} U_{\text{T}}}{\kappa V_{\text{inj}}}}, \tag{8.7}$$

and $I_{\text{tun0}} = I_{\text{inj0}}$.

The dynamics of the floating-gate voltage variation can be described as

$$\frac{d\Delta V_{\text{fg}}}{dt} = \frac{1}{C_{\text{T}}} \cdot (I_{\text{tun}} - I_{\text{inj}}), \tag{8.8}$$

where C_{T} is the total capacitance seen from the floating gate. The time derivative of the floating-gate voltage can also be related to the time derivative of the weight as

$$\frac{d\Delta V_{\text{fg}}}{dt} = -\frac{U_{\text{T}}}{\kappa W} \cdot \frac{dW}{dt}. \tag{8.9}$$

If (8.6) and (8.7) are substituted into (8.8) and if we equate (8.8) and (8.9), the weight dynamics can be modeled as

$$\frac{dW}{dt} = -\frac{\kappa I_{\text{tun0}}}{U_{\text{T}} C_{\text{T}}} \left(W^{1+\frac{U_{\text{T}}}{\kappa V_{\text{x}}}} - W^{1+\alpha-\frac{A_{\text{v}} U_{\text{T}}}{\kappa V_{\text{inj}}}} \right). \tag{8.10}$$

It can be shown that the equilibrium point of $W = 1$ in (8.10) is stable. Therefore, when tunneling and injection mechanisms are both activated, the currents through M_1 and M_2 will converge to half of I_{b} and the floating-gate voltages of $V_{\text{fg,1}}$ and $V_{\text{fg,2}}$ will converge to a same value.

8.2 Center-Adaptive Bump Circuits

Based on the circuit in Figure 8.1b, an adaptive vector quantizer made from an array of adaptive bump circuits was proposed in [45]. The schematic of the adaptive bump circuit is

shown in Figure 8.2. When the "Adapt" control line is low, M_{1-7} form the simple floating-gate bump circuit as shown in Figure 6.2. The output current computes the similarity between two floating-gate voltages and is a function of

$$\begin{aligned} |V_{\mathrm{fg},1} - V_{\mathrm{fg},2}| &= |(V_{\mathrm{in}1} + V_{\mathrm{Q}1}) - (V_{\mathrm{in}2} + V_{\mathrm{Q}2})| \\ &= |(V_{\mathrm{in}1} - V_{\mathrm{in}2}) - (V_{\mathrm{Q}2} - V_{\mathrm{Q}1})| \\ &= |\Delta V_{\mathrm{in}} - \Delta V_{\mathrm{Q}}|, \end{aligned} \tag{8.11}$$

where $\Delta V_{\mathrm{in}} = V_{\mathrm{in}1} - V_{\mathrm{in}2}$ is the input signal and $\Delta V_{\mathrm{Q}} = V_{\mathrm{Q}2} - V_{\mathrm{Q}1}$ is the template information stored on the floating gates. When the "Adapt" control line is high, the floating-gate voltages will be adapted to a same value by tunneling and injection currents. As the floating-gate voltages move toward each other, the template information ΔV_{Q} tends to move toward the input signal ΔV_{in} and the output current tends to reach the maximum value. If the adaptation rate is slow compared with the input signal changes, the template information will move around the mean of input signals.

The block diagram of a resultant adaptive vector quantizer is shown in Figure 8.3. The common-mode feedback block is shared by the bump cells in the same column. A

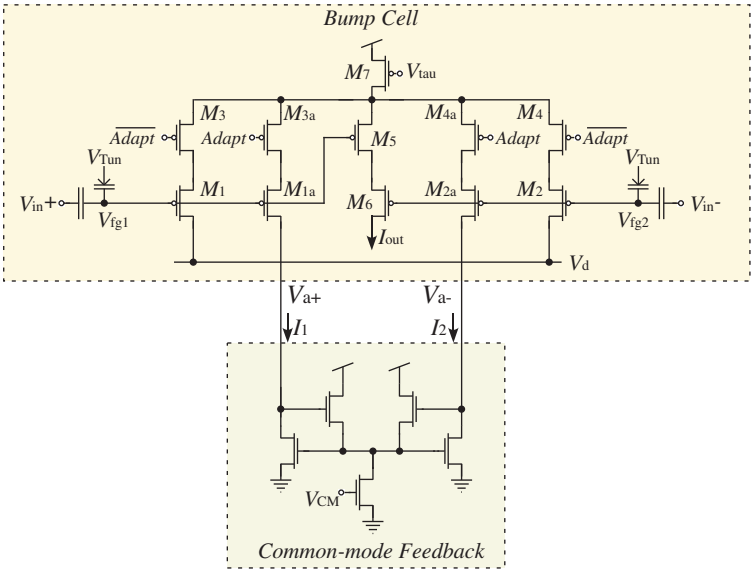


Figure 8.2. An adaptive floating-gate bump circuit.

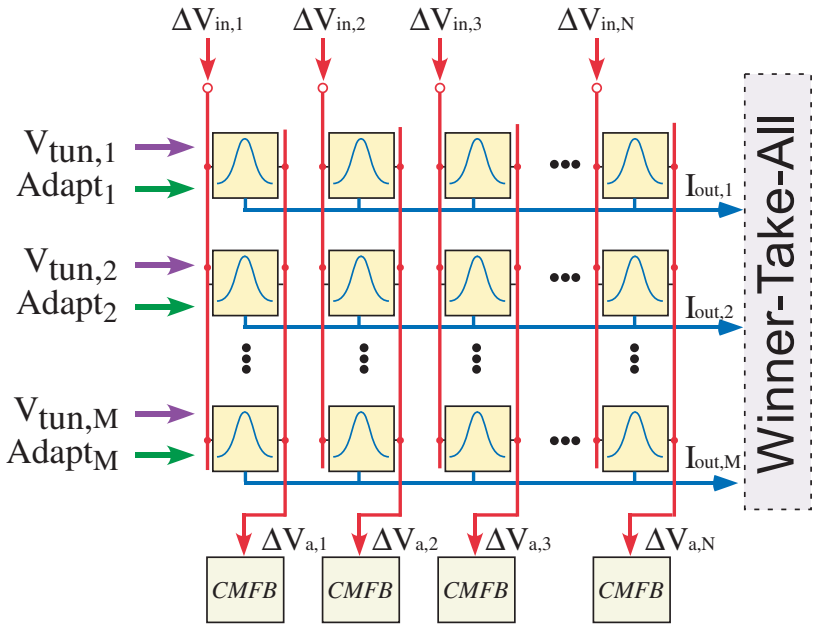


Figure 8.3. An center-adaptive vector quantizer.

winner-take-all circuit decides which row is most similar to the input and then only activates the adaptation mechanisms at that row. The template at the winning row moves toward the input while the templates in other rows are unchanged. Therefore the adaptive vector quantizer performs competitive learning for clustering. A similar approach to perform competitive learning is proposed in [46]. The schematic of an automaximizing bump circuit is shown in Figure 8.4.

8.3 Center-and-Width Adaptive Bump Circuit

The centers of the transfer curves in the adaptive bump circuits shown in Figures 8.2 and 8.4 adapt to the input signal. However, the width of their transfer curve is fixed. Therefore, the resultant adaptive vector quantizers can only model the means but not the variances of the input clusters. To learn both clusters' means and variances, the width of the bump circuit transfer curve needs to be adapted too. In this research, a novel adaptive bump circuit is proposed based on the floating-gate bump circuit presented in Chapter 6 and the schematic is shown in Figure 8.5.

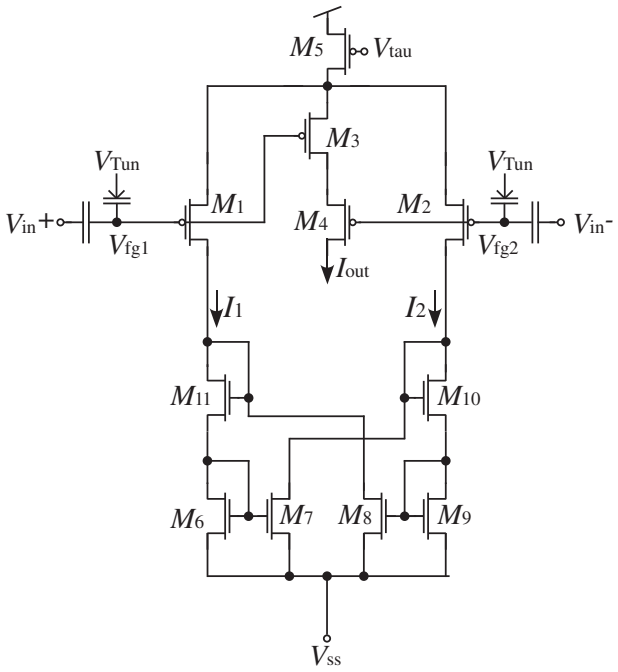


Figure 8.4. An Automaximizing bump circuit.

The center of the transfer curve is controlled by the differential charge and the width is controlled by the common-mode charge of M_1 and M_2. If the adaptation is activated, M_{1-4a} form a differential pair connected to a common-mode feedback circuit. Two negative charge pumps generate negative drain voltages for M_{1b} and M_{2b} that are linearly dependent on the voltages of V_3 and V_4. The indirect injection puts electrons on the floating gates. The floating-gate voltages will adapt to a same value and the center of the transfer curve moves toward the input signal. The amount of the equilibrium common-mode charge on the floating gates sets the width of the transfer curve and is controlled by the tail current of the differential pair. If the distance between the input and the template is far, $|V_1 - V_2|$ is large and the output current of the anti-bump circuit in Figure 8.5 is high. This output current is duplicated using pFET current mirrors to M_{3-4a} and M_{3-4b} as the tail current for adaptation. Therefore, the further between the input and the template, the higher the tail current; as a result, more electrons are put on the floating gates and the bump-shaped transfer curve becomes wider. In this case, if the width adaptation rate is lower than the center adaptation rate, the transfer function of this adaptive bump circuit can approximate

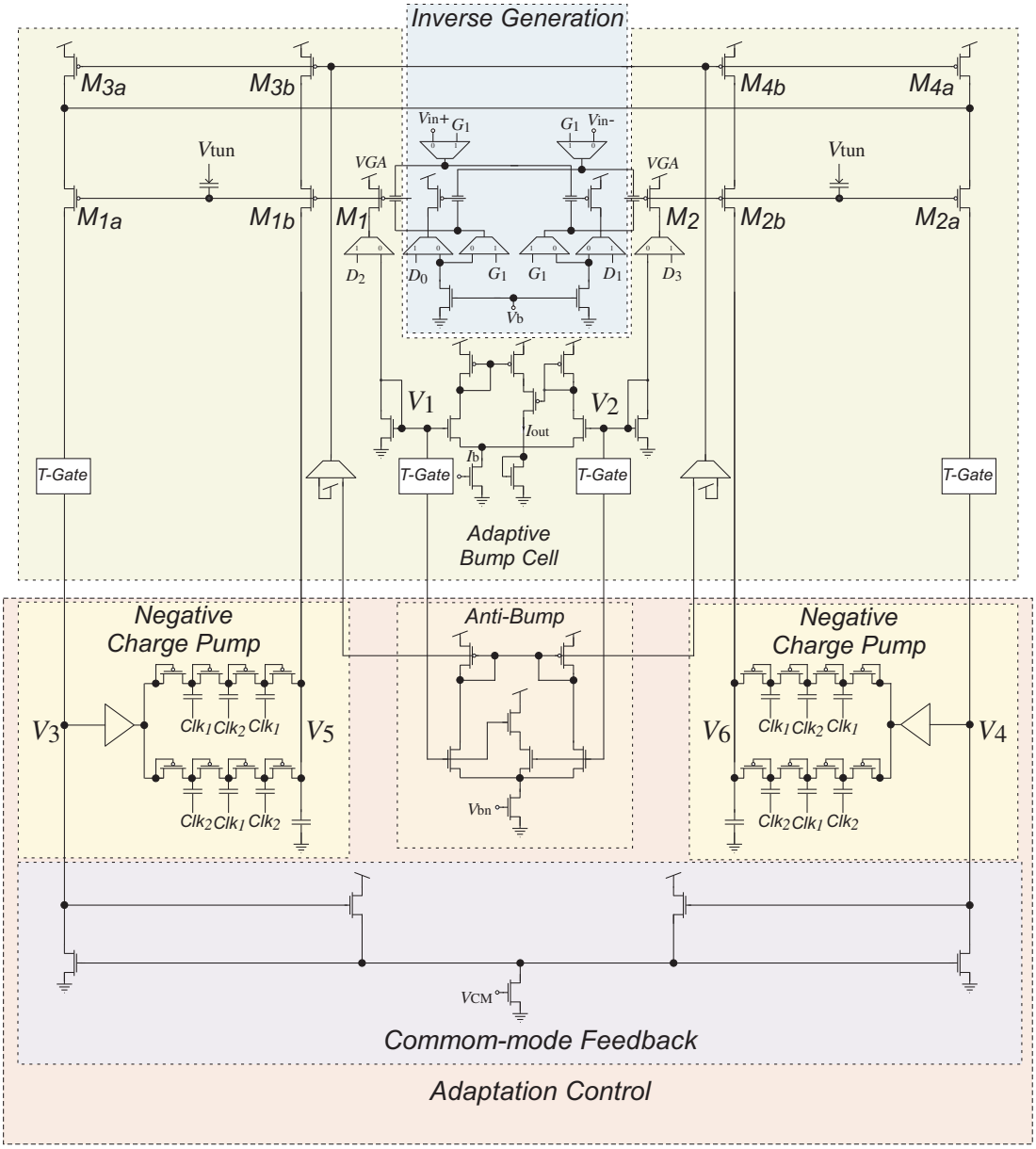


Figure 8.5. A center-and-width adaptive bump circuit.

both the mean and the variance of the input distribution.

The adaptation can be deactivated by turning off the currents through M_{3-4a} and M_{3-4b} to shut down the injection and reducing the tunneling voltage V_{tun} to stop the tunneling. Since the tunneling voltage is much higher than the process nominal voltage, a special tunneling-select circuit is adopted from [46, 47] and its schematic is shown in Figure 8.6a. A high voltage nFET using n-well as the drain terminal is used to sustain the high voltage

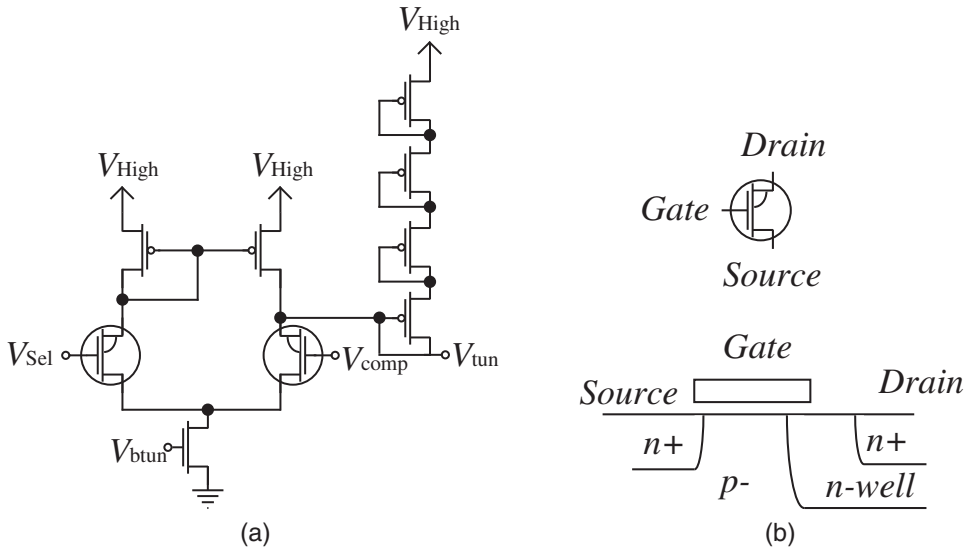


Figure 8.6. The schematics of a tunneling select circuit and the structure of a high-voltage nFET.
(a) Tunneling-select circuit
(b) High voltage nFET

across the transistor. The structure of the high voltage nFET is shown in Figure 8.6b.

In this adaptation approach, the operation of ramping up the V_{DD} for injection is avoided because of the built-in negative charge pumps. The adaptation rate can be adjusted by changing the V_{tun} voltage. When the adaptation is deactivated, the floating-gate bump circuit operates in the same way as presented in Chapter 6. This adaptation structure can also be expanded to implement an analog memory where the input signals can be stored as the floating-gate charges.

8.4 Adaptive Vector Quantizer for Clustering

The architecture of the resultant adaptive vector quantizer using the adaptive bump circuit in Figure 8.5 is shown in Figure 8.7. The "Inverse Generation" and the "Adaptation Control" blocks are shared by the adaptive bump cells in the same column. If a row of the bump circuits is selected to adapt, the "Tunneling Select" block applies a high voltage to the tunneling line and the "Adaptation Control" is connected to that row through transmission gates and multiplexers. With the winner-take-all circuit, the adaptive vector quantizer can perform competitive learning for clustering. Both the means and the variances of clusters

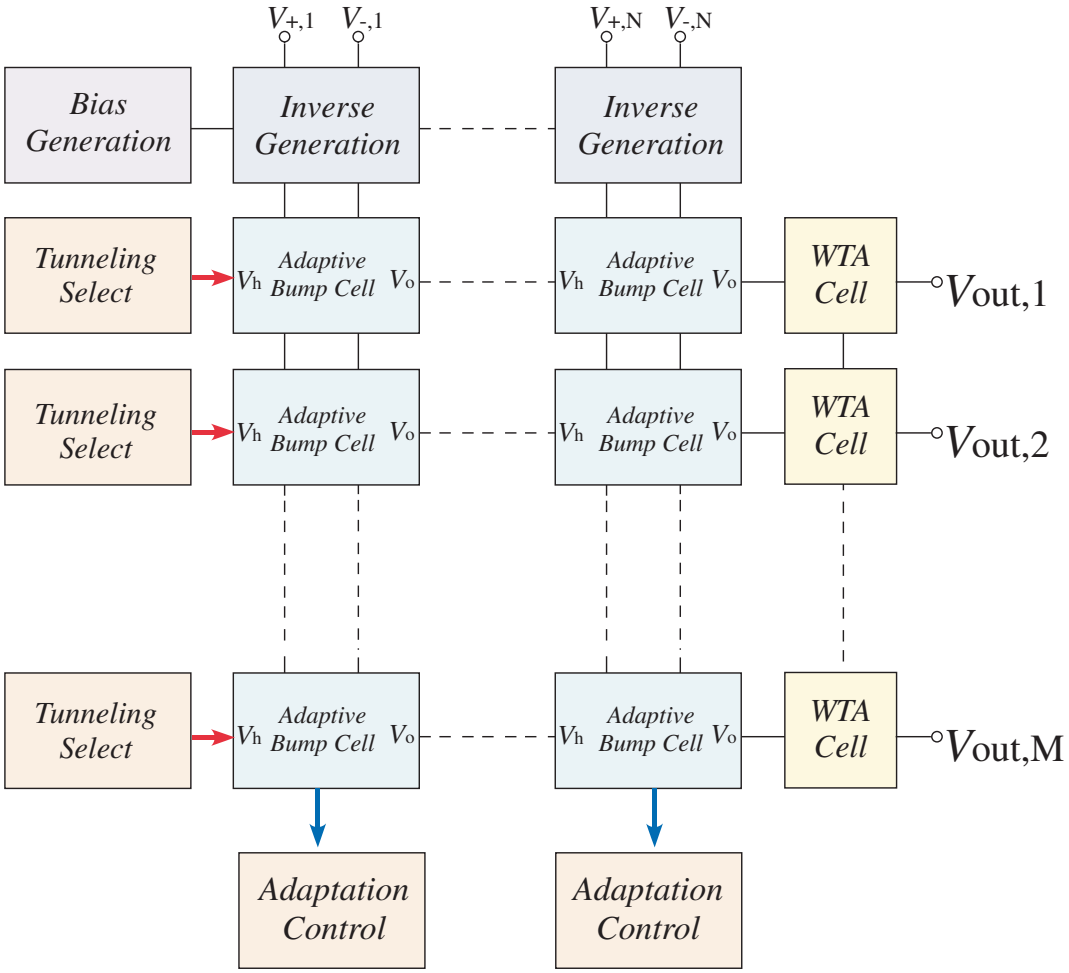


Figure 8.7. An adaptive vector quantizer.

can also be learned by the adaptive bump circuits.

CHAPTER 9

ANALOG SUPPORT VECTOR MACHINE

An analog VLSI implementation of a binary classification algorithm, the support vector machine, is presented in this chapter. A projection neural network that solves the intrinsic optimization problem of the support vector machine was derived based on the work of [48]. The proposed projection neural network derived in this chapter is consistent with the results in [49]. In the hardware implementation, the kernel function can be realized by the bump circuit introduced in Chapter 6. Other circuits include simple current mirrors and log-domain filters. Neither resistors nor operational amplifiers are employed in this implementation. Therefore, compared with [49, 50], this work is more suitable for large-scale analog VLSI implementations. In addition to the block diagram, the complete system and circuits are verified on the transistor level using a SPICE simulator. The same approach can also be applied to support vector regression. With these analog signal processing techniques, a low-power adaptive analog system having the capability of learning, classifying, and regressing becomes feasible.

9.1 Analog for Programming Problems

Support vector machine (SVM) techniques for classification and regression provide powerful tools for learning models that generalize well even in sparse, high dimensional settings [51]. However, this algorithm is considered to be one of the most time-consuming machine learning algorithms because of its intrinsic constrained quadratic programming (CQP) problem. The computational load for SVM learning using a software solver is dominated by the kernel function calculation and the iterations of solving both the primal and the dual variables from a semi-definite equation system [52]. A couple of algorithms are proposed in [53] to speed up the computation of SVM training using a computer. These methods are focused on decomposing the main CQP problem into smaller ones so that

the computation is affordable using limited hardware resources. In many applications, the demands of real-time data processing are often needed.

To improve the computational efficiency of the SVM algorithm, several customized hardware systems have been developed [54–56]. In [54], a digital architecture and an FPGA implementation for SVM have been proposed targeting a channel equalization problem. A mixed-signal processor, the "Kerneltron," proposed in [55] facilitates the computation of vector-matrix-multiplication operations in the SVM algorithm for streaming video applications. In [56], an analog SVM classifier has been used for low-power biometric signature verification. Although analog signal processing techniques have been used in [55,56] leading to significant power savings, these works do not focus on the implementation of SVM learning.

Tank and Hopfield's work [57] has been the basis of many ideas of developing neural networks to solve nonlinear programming (NP) problems. Subsequently, Kennedy and Chua [58] developed networks to solve constrained nonlinear programming (CNP) problems by introducing penalty functions. Their proposed canonical circuit topology has been applied to the first analog circuit implementation of SVM learning [59] in 1998. However, this network does not provide the exact solution unless the penalty coefficient is infinite. Some approaches based on Lagrange multipliers have been proposed [60–63]. In 1996, a projection neural network was proposed by Wu and Xia for solving linear and quadratic programming problems [64]. This projection neural network has been applied to SVM learning in 2001 [65]. Later, Xia and Wang proposed a generic methodology for designing recurrent neural networks for solving CNP problems [66, 67]. Based on [66], several neural networks [68, 69] have been proposed to implement SVM learning. A survey on analog VLSI implementations of SVM [70] provides convenient guide to this research field.

9.2 Support Vector Machine For Classification

Given a training set of $\{(\boldsymbol{x}_i, y_i), i = 1, ..., N\}$ drawn i.i.d. from a distribution $P(\boldsymbol{x}, y)$, where $\boldsymbol{x}_i \in \mathbf{X} \subseteq \mathbb{R}^n$ is the input pattern of the *i-th* training sample with the class labeled as $y_i \in \{+1, -1\}$, the objective of the support vector machine learning algorithm is to obtain the optimal classifier that can approximate the distribution, P. To form a nonlinear classifier, the input space $\mathbf{X}$ can be expanded to a high-dimensional feature space $\mathbf{Z} \subseteq \mathbb{R}^m$ by a mapping function $\varphi : \mathbf{X} \to \mathbf{Z}$ with $m \geq n$. The resultant classifier can be written as

$$h_{\mathbf{w},b}(\boldsymbol{x}) = sign\left[\mathbf{w} \cdot \varphi(\boldsymbol{x}) + b\right], \tag{9.1}$$

where $\mathbf{w} \in \mathbb{R}^m$ is the normal vector and b is the offset of the hyperplane; $sign[\cdot]$ is the bipolar sign function.

The SVM learning algorithm maximizes the margin and minimizes the training errors. The margin is the reciprocal of the length of $\mathbf{w}$. Maximizing the margin is equivalent to minimizing the value of $\mathbf{w}^{\mathrm{T}}\mathbf{w}$. If the error of the *i-th* training sample is denoted by ξ_i, the SVM learning algorithm can be described as

$$\min_{\mathbf{w},b,\xi_i} \left[\frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C\sum_{i=1}^{N}\xi_i\right] \tag{9.2}$$

subject to

$$\begin{cases} y_i\left[\mathbf{w} \cdot \varphi(\boldsymbol{x}_i) + b\right] \geq 1 - \xi_i, & i = 1, ..., N \\ \xi_i \geq 0, & i = 1, ..., N, \end{cases} \tag{9.3}$$

where the constant C is a coefficient used to set the tradeoff between the margin and the total error.

By using Lagrange multipliers, equations of the *Primal* programming problem, (9.2) and (9.3), can be transformed into its *Dual* form that can be written as

$$\min_{\boldsymbol{\alpha}} \left[\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} y_i y_j \alpha_i \alpha_j \varphi(\boldsymbol{x}_i) \cdot \varphi(\boldsymbol{x}_j) - \sum_{i=1}^{N}\alpha_i\right] \tag{9.4}$$

subject to

$$\begin{cases} 0 \leq \alpha_i \leq C, & i = 1, ..., N \\ \sum_{i=1}^{N} \alpha_i y_i = 0, \end{cases} \tag{9.5}$$

where α_i is the Lagrange multiplier of the *i-th* training sample. The normal vector of the hyperplane can be expressed as

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \varphi(\boldsymbol{x}_i), \tag{9.6}$$

and the classifier equation becomes

$$h_{\alpha,b}(\boldsymbol{x}) = sign\left[\sum_{i=1}^{N} \alpha_i y_i \varphi(\boldsymbol{x}_i) \cdot \varphi(\boldsymbol{x}) + b\right]. \tag{9.7}$$

From (9.4) and (9.7), if the inner product of $\varphi(\boldsymbol{x}_i)$ and $\varphi(\boldsymbol{x}_j)$ is replaced by a kernel function

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \varphi(\boldsymbol{x}_i) \cdot \varphi(\boldsymbol{x}_j), \tag{9.8}$$

the calculation of the exact value of each $\varphi(\boldsymbol{x}_i)$ can be avoided and the computational complexity can be significantly reduced. This is known as the "kernel trick". Several commonly used kernel functions include

$$\begin{aligned} linear: &\quad k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i \cdot \boldsymbol{x}_j \\ polynomial: &\quad k(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i \cdot \boldsymbol{x}_j + 1)^p \\ Guassian: &\quad k(\boldsymbol{x}_i, \boldsymbol{x}_j) = e^{-\gamma \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}. \end{aligned} \tag{9.9}$$

By using the kernel functions, the *Dual* CQP problem can be rewritten in a matrix form as

$$\min_{\boldsymbol{\alpha}} \left[\frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}} \boldsymbol{Q} \boldsymbol{\alpha} - \mathbf{e}^{\mathrm{T}} \boldsymbol{\alpha}\right] \tag{9.10}$$

subject to

$$\begin{cases} 0 \le \boldsymbol{\alpha} \le C \\ \boldsymbol{\alpha}^{\mathrm{T}} \mathbf{y} = 0, \end{cases} \tag{9.11}$$

where $\boldsymbol{\alpha}$ is a Lagrange multiplier vector and $\boldsymbol{e} = (1, ..., 1)^{\mathrm{T}}$. The element in $\boldsymbol{Q}$ can be expressed as

$$q_{i,j} = y_i y_j k(\boldsymbol{x}_i, \boldsymbol{x}_j). \tag{9.12}$$

The classifier becomes

$$h_{\alpha,b}(\boldsymbol{x}) = sign\left[\sum_{i=1}^{N} \alpha_i y_i k(\boldsymbol{x}_i, \boldsymbol{x}) + b\right]. \tag{9.13}$$

The objective of the SVM learning algorithm is to find the optimal values, $\boldsymbol{\alpha}^*$ and b^*, of the CQP problem.

9.3 Projection Neural Network for SVM Learning

To implement the SVM learning algorithm using analog circuits, a set of ordinary differential equations (ODEs) in the form of

$$\dot{\boldsymbol{u}} = \boldsymbol{\Lambda} \boldsymbol{F}(\boldsymbol{u}) \tag{9.14}$$

needs to be derived and the equilibrium point of this dynamic system provides the optimal solution to the CQP problem.

9.3.1 Relevant theorems

To derive the set of ODEs for solving the SVM learning problem, several relevant theorems are described first in this subsection.

Theorem 1 (Kuhn-Tucker Saddle-point Condition [71]) *Assume an optimization problem of the form*

$$\min_{\boldsymbol{u}} f(\boldsymbol{u}) \tag{9.15}$$

subject to

$$\begin{cases} g_i(\boldsymbol{u}) \leq 0, & i = 1, ..., m \\ h_j(\boldsymbol{u}) = 0, & j = 1, ..., p, \end{cases} \tag{9.16}$$

where $f, g_i, h_j : \mathbb{R}^n \to \mathbb{R}$ *for* $i \in [1, ..., m]$, $j \in [1, ..., p]$ *are arbitrary functions, and a Lagrangian*

$$L(\boldsymbol{u}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = f(\boldsymbol{u}) + \sum_{i=1}^{m} \alpha_i g_i(\boldsymbol{u}) + \sum_{j=1}^{p} \beta_j h_j(\boldsymbol{u})$$

where $\alpha_i \geq 0$ *and* $\beta_j \in \mathbb{R}$*. If a set of variables* $(\boldsymbol{u}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$*, for* $\boldsymbol{u}^* \in \mathbb{R}^n$*,* $\boldsymbol{\alpha}^* \in [0, \infty)^m$*, and* $\boldsymbol{\beta}^* \in \mathbb{R}^p$ *exists such that for all* $\boldsymbol{u} \in \mathbb{R}^n$*,* $\boldsymbol{\alpha} \in [0, \infty)^m$*, and* $\boldsymbol{\beta} \in \mathbb{R}^p$*,*

$$L(\boldsymbol{u}^*, \boldsymbol{\alpha}, \boldsymbol{\beta}) \leq L(\boldsymbol{u}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*) \leq L(\boldsymbol{u}, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*), \tag{9.17}$$

then $\boldsymbol{u}^*$ *is a solution to (9.15) and (9.16).*

In [71], it is also shown that with the additional assumptions that f and g_i's are convex on the convex set $\boldsymbol{\Omega} \subseteq \mathbb{R}^n$ and that g_i's satisfy some "nice" constraints, which is the case for SVM learning, the saddle point criterion is also a necessary condition for optimality. These "nice" constraints can be either the *Slater's condition*, *Karlin's condition* or *Strict constraint qualification* [72].

According to [73], the problem in the form of (9.17) is equivalent to a linear variational inequality problem, $VI(\boldsymbol{U}, \Omega_0)$, of finding $\boldsymbol{v}^* \in \Omega_0$ satisfying

$$(\boldsymbol{v} - \boldsymbol{v}^*)^{\mathrm{T}} \boldsymbol{U}(\boldsymbol{v}^*) \geq 0, \quad \forall \boldsymbol{v} \in \Omega_0, \tag{9.18}$$

where Ω_0 is a closed convex set. The solution to this linear variational inequality problem, $VI(\boldsymbol{U}, \Omega_0)$, satisfies the following projection theorem.

Theorem 2 (Projection Theorem [74]) $\boldsymbol{u}^*$ *is a solution to* $VI(\boldsymbol{U}, \Omega_0)$ *if and only if* $\boldsymbol{u}^*$ *satisfies*

$$P_{\Omega_0}(\boldsymbol{u} - \lambda \boldsymbol{U}(\boldsymbol{u})) = \boldsymbol{u}, \tag{9.19}$$

where λ *is any positive constant, and* $P_{\Omega_0}(\boldsymbol{u}) : \mathbb{R}^n \to \Omega_0$ *is a projection operator that is defined as*

$$P_{\Omega_0}(\boldsymbol{u}) = \arg\min_{\boldsymbol{v} \in \Omega_0} \|\boldsymbol{u} - \boldsymbol{v}\|.$$

If $\boldsymbol{U}(\boldsymbol{u})$ can be written in the form of $\boldsymbol{M}\boldsymbol{u} + \boldsymbol{q}$, where $\boldsymbol{M}$ is an $n \times n$ positive-definite matrix, $\boldsymbol{q} \in \mathbb{R}^n$, the equilibrium point of the following dynamic system

$$\frac{d\boldsymbol{u}}{dt} = \lambda \left\{ P_{\Omega_0}(\boldsymbol{u} - (\boldsymbol{M}\boldsymbol{u} + \boldsymbol{q})) - \boldsymbol{u} \right\}, \tag{9.20}$$

is the solution to the linear variational inequality

$$(\boldsymbol{u} - \boldsymbol{u}^*)^{\mathrm{T}}(\boldsymbol{M}\boldsymbol{u}^* + \boldsymbol{q}) \geq 0, \quad \forall \boldsymbol{u} \in \Omega_0. \tag{9.21}$$

If $\boldsymbol{u}_0 \in \Omega_0$, this dynamic system is globally asymptotically stable [48].

9.3.2 The projection neural network for SVM

From (9.2) and (9.3) in the *Primal* CQP, the Lagrangian can be rearranged as

$$W(\boldsymbol{\alpha}, b) = \frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{\alpha} - \boldsymbol{e}^{\mathrm{T}}\boldsymbol{\alpha} + b \cdot \left(\boldsymbol{y}^{\mathrm{T}}\boldsymbol{\alpha}\right). \tag{9.22}$$

It is noticeable that $\boldsymbol{\alpha}$ is the dual variable and b is the primal variable. According to the saddlepoint condition, the following inequalities

$$W(\boldsymbol{\alpha}^*, b) \leq W(\boldsymbol{\alpha}^*, b^*) \leq W(\boldsymbol{\alpha}, b^*) \tag{9.23}$$

hold, where $0 \leq \boldsymbol{\alpha} \leq C$, $b \in \mathbb{R}^n$. $\boldsymbol{\alpha}^*$ and b^* are the optimal values.

Based on [73], the saddle-point problem of (9.23) is equivalent to a variational inequality problem, $VI(\boldsymbol{U}, \Omega_0)$, with $\boldsymbol{u} = (\boldsymbol{\alpha}^{\mathrm{T}}, b)^{\mathrm{T}}$,

$$\boldsymbol{U}(\boldsymbol{u}) = \boldsymbol{U}_{\alpha,b}(\boldsymbol{\alpha}, b) = (\nabla_{\alpha} W(\boldsymbol{\alpha}, b), -\nabla_b W(\boldsymbol{\alpha}, b)) \tag{9.24}$$

and $\Omega_0 = \left\{\boldsymbol{u} = (\boldsymbol{\alpha}^{\mathrm{T}}, b)^{\mathrm{T}} |\ 0 \leq \boldsymbol{\alpha} \leq C,\ b \in \mathbb{R}\right\}$. After substituting (9.22) into (9.24), the function, $\boldsymbol{U}(\boldsymbol{u})$, can be written as

$$\begin{aligned} \boldsymbol{U}(\boldsymbol{u}) = \quad & \boldsymbol{U}_{\alpha,b}(\boldsymbol{\alpha}, b) = \begin{pmatrix} \boldsymbol{Q} & \boldsymbol{y} \\ -\boldsymbol{y}^{\mathrm{T}} & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha} \\ b \end{pmatrix} + \begin{pmatrix} -\boldsymbol{e} \\ 0 \end{pmatrix} \\ = \quad & \boldsymbol{M}\boldsymbol{u} + \boldsymbol{q}. \end{aligned} \tag{9.25}$$

It is easy to show that if $\boldsymbol{Q}$ is positive-definite, $\boldsymbol{M}$ is also positive-definite. Based on (9.20) and (9.21), the differential equations of the projection neural network for SVM learning can be derived as

$$\frac{d\boldsymbol{\alpha}}{dt} = \lambda \left\{P_{\Omega_0}(\boldsymbol{\alpha} - \boldsymbol{Q}\boldsymbol{\alpha} - b\boldsymbol{y} + \boldsymbol{e}) - \boldsymbol{\alpha}\right\} \tag{9.26}$$

$$\frac{db}{dt} = \lambda \left(\boldsymbol{y}^{\mathrm{T}}\boldsymbol{\alpha}\right), \tag{9.27}$$

where λ is a positive number, and

$$\Omega_0 = \left\{ \boldsymbol{u} = (\boldsymbol{\alpha}^{\mathrm{T}}, b)^{\mathrm{T}} | \; 0 \leq \boldsymbol{\alpha} \leq C, \; b \in \mathbb{R} \right\}. \tag{9.28}$$

The element in $\boldsymbol{Q}$ is given as

$$q_{i,j} = y_i y_j k\left(\boldsymbol{x}_i . \boldsymbol{x}_j\right), \tag{9.29}$$

Compared with previous works [59, 65, 68], this projection neural network provides the exact solution, unlike the penalty function approaches [59]. Besides, the number of the state variables of this neural network is minimum. This projection neural network for SVM learning is simpler than those previous works.

Although in different forms, the derived equations, from (9.26) to (9.29), are consistent with the equations proposed in [49]. The equations derived in [49] are given as

$$\frac{d\boldsymbol{\alpha}}{dt} = \lambda \left\{ P_{\Omega_1}(\boldsymbol{\alpha} - \widehat{\boldsymbol{Q}}\boldsymbol{\alpha} - b\boldsymbol{e} + \boldsymbol{y}) - \boldsymbol{\alpha} \right\} \tag{9.30}$$

$$\frac{db}{dt} = -\lambda \left(\boldsymbol{e}^{\mathrm{T}} \boldsymbol{\alpha} \right), \tag{9.31}$$

where λ is a positive number, and the element of $\widehat{\boldsymbol{Q}}$ is

$$\hat{q}_{i,j} = k\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right). \tag{9.32}$$

The convex set Ω_1 in (9.30) is given as

$$\Omega_1 = \left\{ \boldsymbol{u} = (\boldsymbol{\alpha}^{\mathrm{T}}, b)^{\mathrm{T}} | \; \boldsymbol{d} \leq \boldsymbol{\alpha} \leq \boldsymbol{h}, \; b \in R \right\} \tag{9.33}$$

where

$$d_i = \begin{cases} -C & y_i = -1 \\ 0 & y_i = 1 \end{cases} \tag{9.34}$$

and

$$h_i = \begin{cases} C & y_i = 1 \\ 0 & y_i = -1 \end{cases}. \tag{9.35}$$

Unlike the convex set Ω_1 in (9.33), the closed convex set Ω_0 in the proposed ODEs is independent of the class vector, $\boldsymbol{y}$. Therefore, the proposed projection neural network is more suitable for analog VLSI implementation.

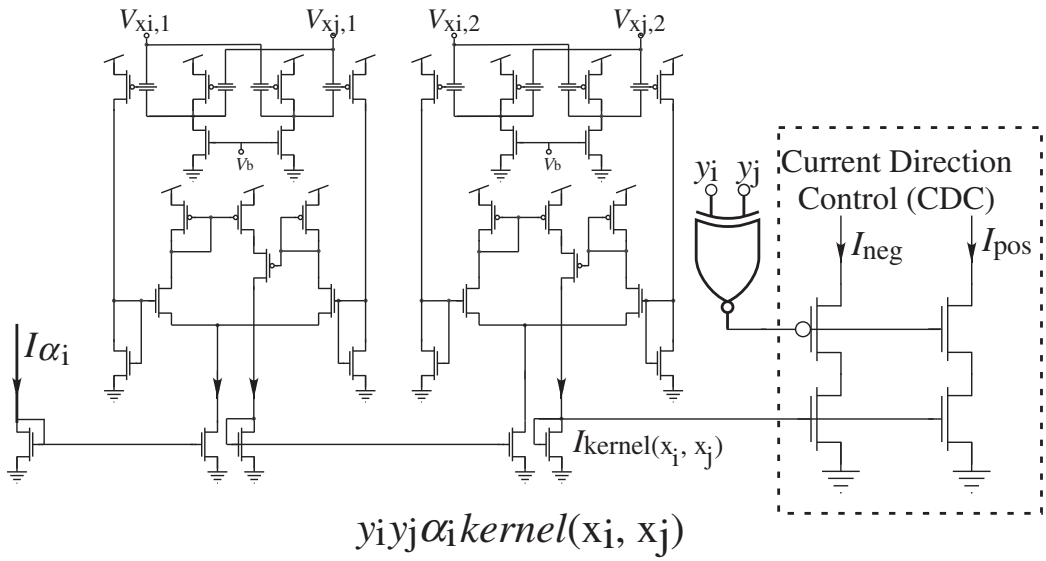


Figure 9.1. The schematic of the *Kernel* block.

9.4 Analog Implementation of The Support Vector Classification

In the analog implementation, the kernel function, $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, is chosen as a radial basis function and is realized by the output current of a proposed floating-gate bump circuit that has been detailed in Chapter 6. A multivariate radial basis function can be implemented by cascading these proposed bump circuits as shown in Figure 6.9. The state variables, α_i's and b, in (9.26) and (9.27) are realized by current signals, I_{α_i}'s, and I_b. The classes, y_i's, are set as digital signals. The circuit used to compute the element in $\boldsymbol{Q\alpha}$ is named as a *Kernel* block. It realizes the term of $y_i y_j \alpha_i k\left(\boldsymbol{x}_i.\boldsymbol{x}_j\right)$. The schematic of the *Kernel* block is shown in Figure 9.1. The tail current of the first stage bump circuit is set to be I_{α_i} and the output current of the bump circuit in the final stage is $\alpha_i k\left(\boldsymbol{x}_i.\boldsymbol{x}_j\right)$. A digital XNOR-gate computes the polarity of the output current and determines whether the output current will contribute to the positive current or the negative current.

It is straightforward to implement the terms of $b\boldsymbol{y}$ and $\boldsymbol{y}^T\boldsymbol{\alpha}$ in (9.26) and (9.27) using the current direction control (CDC) block shown in Figure 9.1. The components of the positive and the negative currents are summed up individually using KCL. The final output current is realized by subtracting the negative current from the positive current using a current

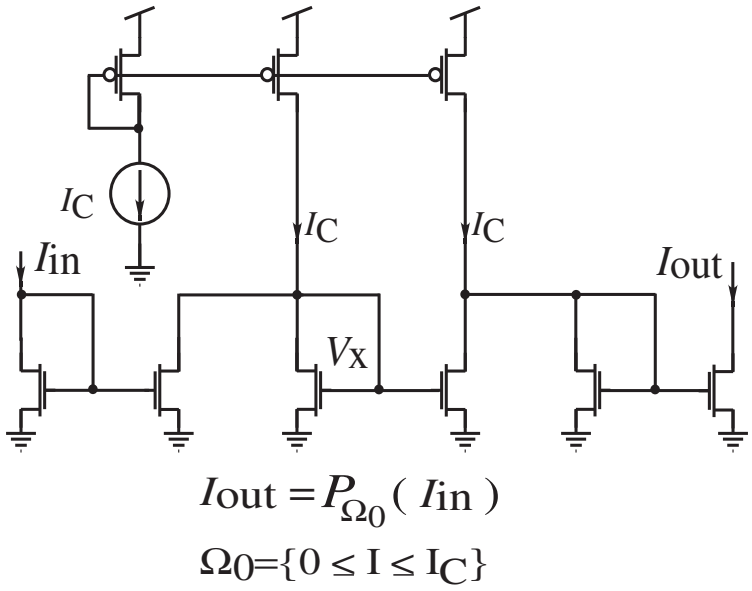


Figure 9.2. The schematic of the *Project* block.

mirror.

A current-mirror-based circuit, named *Project* block, is used to implement the projection operator, P_{Ω_0}, in (9.26). The schematic of the *Project* block is shown in Figure 9.4. If $I_{\text{in}} < I_{\text{C}}$, the output current is equal to the input current. In the case of $I_{\text{in}} > I_{\text{C}}$, the node V_{X} will be pull to ground and the output current will be clamped at I_{C}.

A translinear low-pass filter (LPF) and a translinear integrator circuits complete the computation in (9.26) and (9.27), respectively. The schematics of these two blocks are shown in Figure 9.3 where the transistors operate in the subthreshold region. Because the sign of b can be bipolar and the currents in the translinear circuits need to be positive, an offset current is added to keep the output current in the *Integrator* block positive. The offset current is then subtracted before the *Project* block.

The architecture of the resultant learner for support vector classification is shown in Figure 9.4a. Four training samples are used as illustration. A 4×4 *Kernel* blocks matrix calculates the vector-matrix-multiplication terms of $\boldsymbol{Q\alpha}$. The currents are summed up in rows and are projected to the close convex set, Ω_0 in the *Project* blocks. The *LPF* blocks perform the first-order low-pass filter and generate the state variables α_i's. As to the offset,

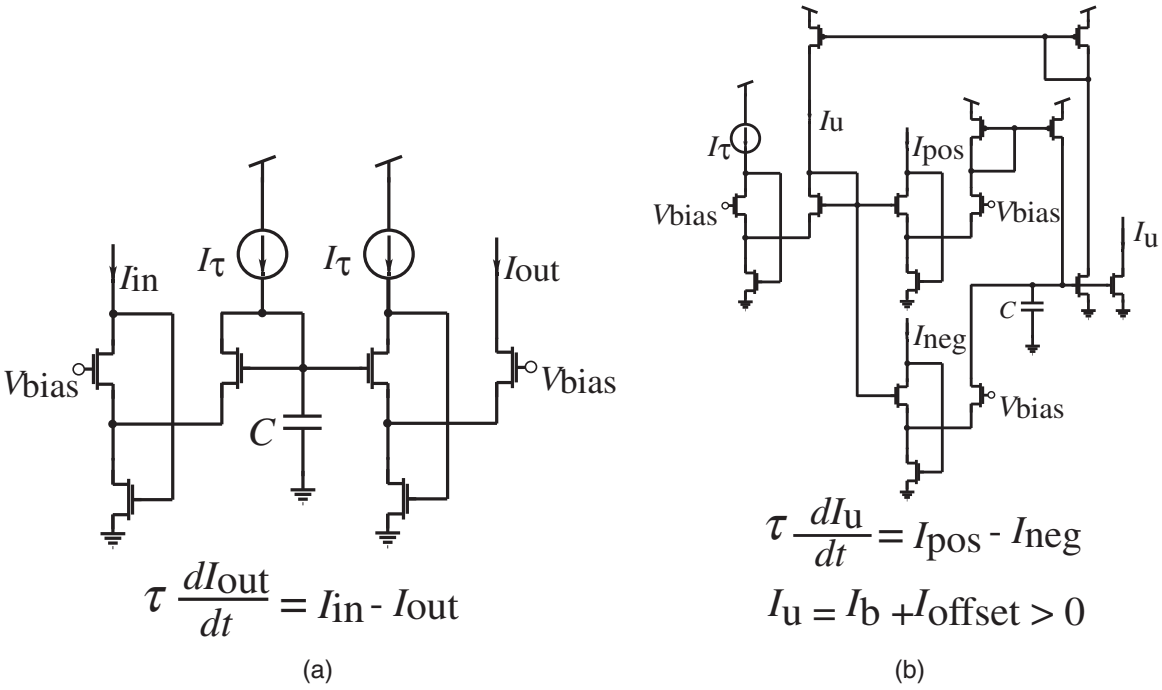


Figure 9.3. The schematics of the *LPF* and the *Integrator* blocks.
(a) The *LPF* block
(b) The *Integrator* block

b, the current direction control (CDC) blocks determine the sign of the current contribution of α_i. Then these currents are summed up and integrated in the *Integrator* block. All state variables, α_i's and b, are fed back to complete the learner dynamic system.

The architecture of the classifier is shown in Figure 9.4b. The circuit in the kernel block is the same as the circuit in Figure 9.1 but without the XNOR circuit. The learner output currents and the support vectors are used to determine the class of the input $\boldsymbol{x}$. The sign of the classifier output current and the voltage at the output node indicate the classification result.

In this implementation, the analog signal processing circuits include current mirrors, log-domain filters, and floating-gate bump circuits. If the tunability of the kernel function is not necessary, a simple bump circuit can be used to implement the RBF kernel function and the complexity of the whole system can be further reduced. There are no resistors nor operational amplifiers in the implementation. Besides, most of the transistors operate in

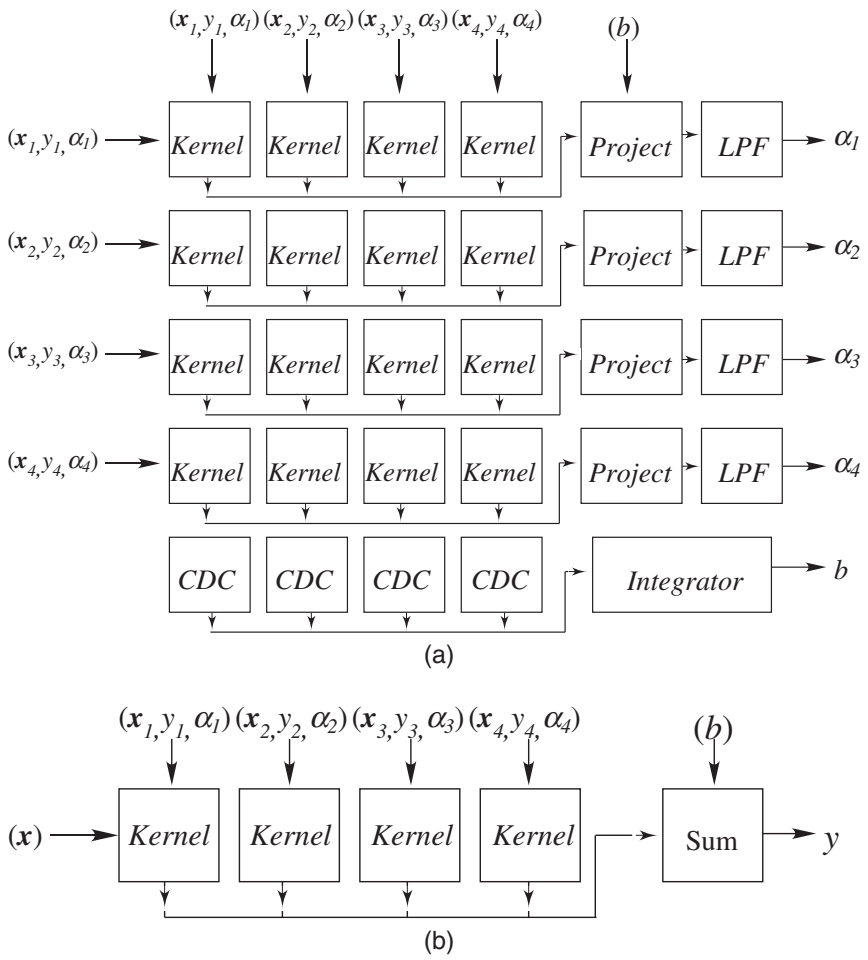


Figure 9.4. The architectures of the analog learner and classifier for SVM classification.
(a) The learner
(b) The classifier

the subthreshold region. Therefore, this approach is highly area and power efficient and is suitable for the implementation of a large-scale projection neural network for support vector classification.

9.5 Simulation Results

Four training samples in a two-dimensional feature space are used to verify the analog VLSI approach to the support vector machine learning and classification. The training sample distribution is shown in Figure 9.5. The plus sign stands for the class of +1 and the circle stands for the class of −1. The dotted region is predicted as −1 using the MATLAB numerical ODE solver. The simulation results of the potential level using MATLAB is

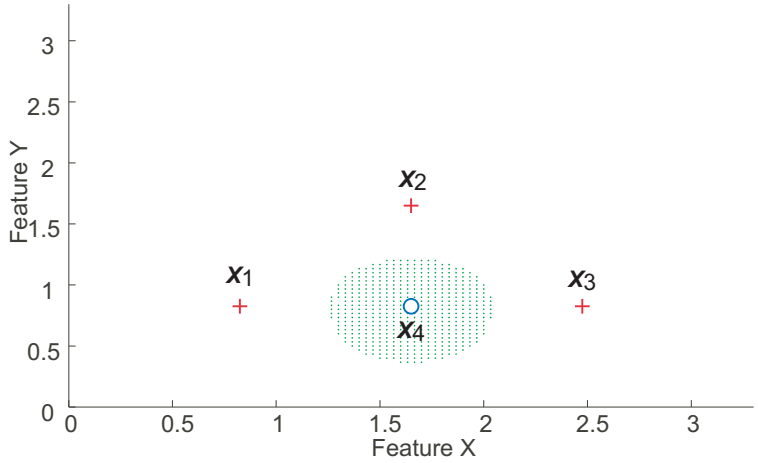


Figure 9.5. The distribution of the training samples.

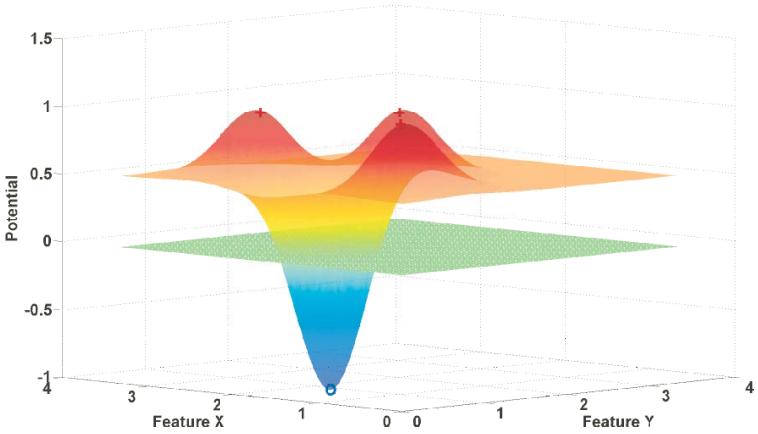


Figure 9.6. The simulation result of the potential level using MATLAB numerical ODE solver.

shown in Figure 9.5 and is consistent with the simulation results using a prevailing SVM program, SVM^{light}, developed at Cornell University.

The same problem is simulated on the transistor level using a SPICE circuit simulator. The simulated transient response of all the state variables is shown in Figure 9.5. The dynamics of the state variables converge to equilibrium values in 0.4 milli-seconds. The learning results are fed to an analog support vector classifier the output current of which indicates the potential level. The simulated output currents using SPICE circuit simulator are shown in Figure 9.8 and the behaviors are in line with those obtained from the digital numerical solvers. In Figure 9.8b, the widths of the bump circuit transfer curves are

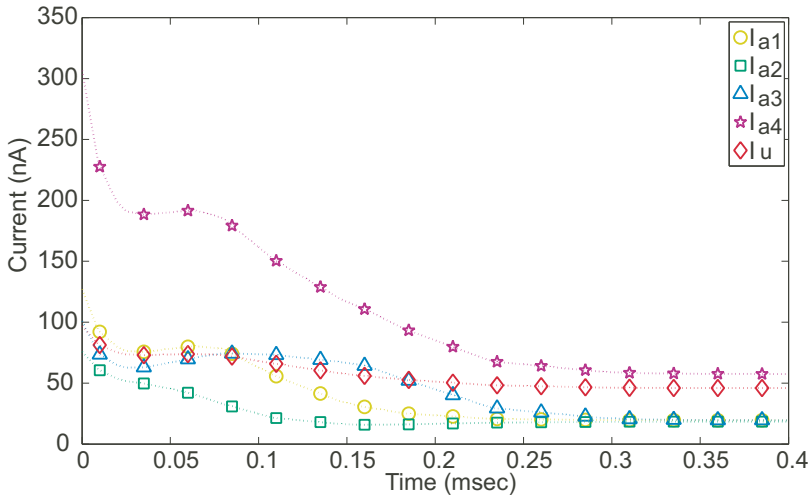


Figure 9.7. The transient response of the state variables simulated in SPICE.

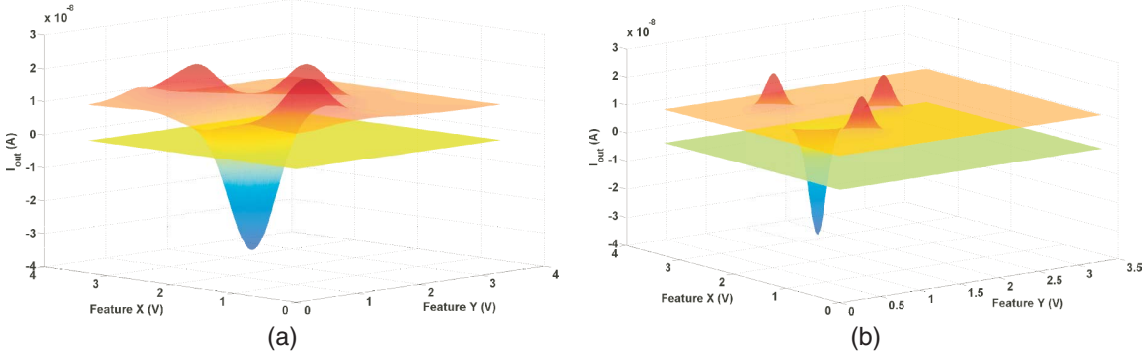


Figure 9.8. The simulated output current using a SPICE circuit simulator.

programmed to be narrower. Being able to change the width of the RBF function in the analog fashion enables the system to fit into different scenarios. Therefore this analog signal processing system can be integrated with sensor interface circuits performing learning and classification tasks, resulting in a smart sensory system without using any power-hungry analog-to-digital converter.

9.6 The Projection Neural Network for Support Vector Regression

In [49], the equations, from (9.30) to (9.35), used to perform support vector classification can also be used for support vector regression (SVR). Although this is not the case for the projection neural network derived in Section 9.3, another projection neural network for

SVR can be derived using the similar approach.

If the cost function of the regression is chosen as

$$cst(\xi_i) = |\xi_i|, \tag{9.36}$$

the optimization problem for regression can be expressed as

$$\min_{\mathbf{w},b,\xi_i} \left[\frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C\sum_{i=1}^{N} \left(cst(\xi_i) + cst(\xi_i^*) \right) \right] \tag{9.37}$$

subject to

$$\begin{cases} \mathbf{w}\cdot\varphi(\boldsymbol{x}_i) + b - y_i \le \xi_i, & i = 1, ..., N \\ y_i - \mathbf{w}\cdot\varphi(\boldsymbol{x}_i) - b \le \xi_i^*, & i = 1, ..., N \\ \xi_i, \xi_i^* \ge 0, & i = 1, ..., N \end{cases}, \tag{9.38}$$

where $y_i \in \mathbb{R}$ is the output value of the *i-th* training sample, φ is the feature expansion mapping function, and ξ_i and ξ_i^* are the error terms from the upper and lower boundaries of the *i-th* training sample.

After applying the "kernel trick" and after some manipulations, the negated Lagrangian can be expressed as

$$W(\boldsymbol{\alpha}, \boldsymbol{\alpha}^*, b) = \frac{1}{2}(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*)^{\mathrm{T}}\widetilde{\boldsymbol{Q}}(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) - \boldsymbol{y}^{\mathrm{T}}(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) + b \cdot \left(\boldsymbol{e}^{\mathrm{T}}(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) \right), \tag{9.39}$$

where $0 \le \boldsymbol{\alpha} \le C$, $0 \le \boldsymbol{\alpha}^* \le C$, and the element in $\widetilde{\boldsymbol{Q}}$ is $\tilde{q}_{i,j} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. By substituting variable $\boldsymbol{a} = (\boldsymbol{\alpha} - \boldsymbol{\alpha}^*)$, we can get:

$$W(\boldsymbol{a}, b) = \frac{1}{2}\boldsymbol{a}^{\mathrm{T}}\widetilde{\boldsymbol{Q}}\boldsymbol{a} - \boldsymbol{y}^{\mathrm{T}}\boldsymbol{a} + b \cdot \left(\boldsymbol{e}^{\mathrm{T}}\boldsymbol{a} \right), \tag{9.40}$$

where $-C \le \boldsymbol{a} \le C$. This CQP problem can be converted into a variational inequality problem using the Kuhn-Tucker Saddle-point Condition [71]. After applying the Projection Theorem [74], the ODEs of the projection neural network for SVR can be derived as

$$\frac{d\boldsymbol{a}}{dt} = \lambda \left\{ P_{\Omega_2}(\boldsymbol{a} - \widetilde{\boldsymbol{Q}}\boldsymbol{a} - b\boldsymbol{e} + \boldsymbol{y}) - \boldsymbol{a} \right\} \tag{9.41}$$

$$\frac{db}{dt} = \lambda \left(\boldsymbol{e}^{\mathrm{T}}\boldsymbol{a} \right), \tag{9.42}$$

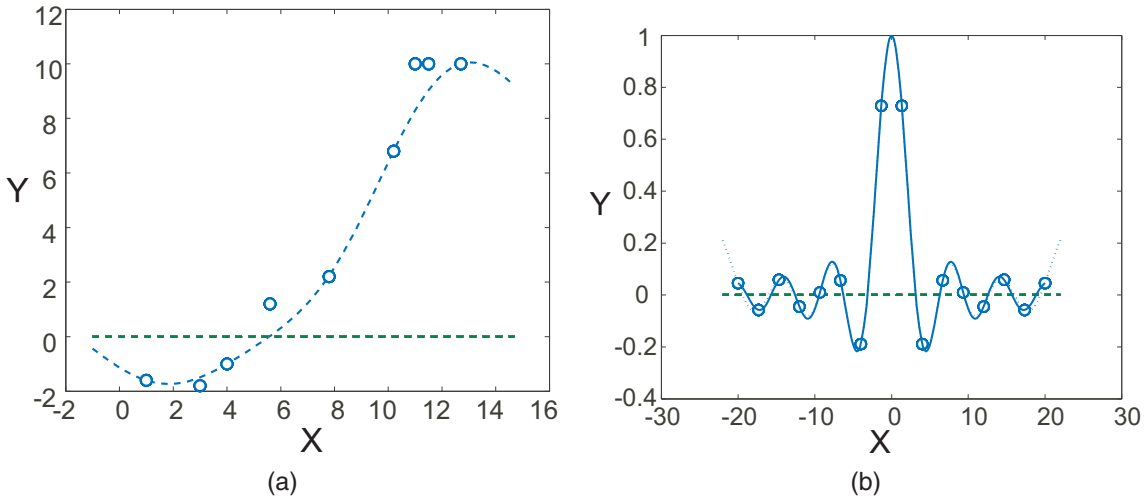


Figure 9.9. The simulated results of the projection neural network for support vector regression.

where $\Omega_2 = \left\{\boldsymbol{u} = (\boldsymbol{a}^{\mathrm{T}}, b)^{\mathrm{T}} | \ -C \leq \boldsymbol{a} \leq C, \ b \in \mathbb{R}\right\}$. The support vector regression estimator can be expressed as

$$g_{\boldsymbol{a},b}(\boldsymbol{x}) = \sum_{i=1}^{N} a_i y_i k(\boldsymbol{x}, \boldsymbol{x}_i) + b \tag{9.43}$$

Compared with (9.26) and (9.27), these equations can be implemented by similar but simpler analog circuits. The support vector machine for regression can also be implemented in analog VLSI.

The simulation results of the projection neural network using the MATLAB numerical ODE solver are shown in Figure 9.9. The circles are the training samples and the dotted lines are the regression results. The solid line in Figure 9.9b is the target function.

9.7 Conclusion

An analog VLSI approach to implementing a projection neural network for solving the optimization problems of the support vector machine algorithm is presented in this chapter. The radial-basis kernel function is realized by floating-gate bump circuits. The analog signal processing circuits use the intrinsic nonlinearity of the silicon devices and do not use any resistors or operational amplifiers. Therefore, this approach is suitable for the analog VLSI implementation of large-scale projection neural networks for support vector machine

algorithms. The transistor-level SPICE simulation verifies the feasibility of this approach.

With these analog signal processing techniques, a low-power adaptive analog system with the capability of learning, classifying and regressing becomes feasible. It can be integrated with the sensor interface circuits to form a highly efficient smart sensory system without employing any analog-to-digital converter.

CHAPTER 10

CONCLUSION

In this chapter, the main contributions and key milestones that have been achieved in this work are summarized.

10.1 Main Contributions

As the emerging sensory microsystems face severer power constraints and as the demands for short design-testing cycles increase, a reconfigurable smart sensory system appears as a useful tool for quickly prototyping innovative sensory systems that require highly power-efficient signal processing. Based on floating-gate technologies and the existing infrastructure of large-scale FPAAs, this work provides significant foundations, mainly in aspects of interface circuits and analog classifiers, for developing a reconfigurable smart sensory chip. A universal sensor interface block has been designed and integrated in a large-scale reconfigurable analog signal processor. A low-power capacitive sensing interface circuit that achieves a large dynamic range has been analyzed, designed, and applied to audio and ultrasonic applications. A compact analog RBF-based classifier that achieves significant power savings compared with digital processors has been designed and has been utilized to perform automatic gender identification experiments. Ground works for the analog VLSI implementation of the support vector machine also have been proposed and verified on the transistor level. This work also sheds light on the development of analog adaptive learning systems.

10.2 Research Summary

A universal sensor interface block has been designed and integrated in a floating-gate based large-scale reconfigurable analog signal processor. Different interface circuits can be synthesized in this block for capacitive sensing, voltage sensing, or current sensing. Therefore,

the resultant chip can interface with different sensors and can implement different analog algorithms for different applications.

A low-power approach to capacitive sensing that can achieve a high signal-to-noise ratio has been proposed, designed, and tested. The circuit is composed of a capacitive feedback charge amplifier and a charge adaptation circuit. Without the adaptation circuit, the charge amplifier only consumes 1 μW and achieves an SNR of 69.34 dB in the audio band. An adaptation scheme using Fowler-Nordheim tunneling and channel hot electron injection mechanisms to stabilize the DC output voltage has been demonstrated. This scheme provides a low-frequency corner at 0.2 Hz. The measured noise spectra show that this slow adaptation does not degrade the circuit performance. The DC path can also be provided by a large feedback resistance without causing extra power consumption [75]. This capacitive feedback charge amplifier also has been used as a receiver circuit for a capacitive micromachined ultrasonic transducer that is designed for forward-looking intravascular ultrasound imaging applications. Compared with conventional approaches, using a charge amplifier to detect capacitance variation avoids the dilemma of sensitivity-bandwidth tradeoff. Pulse-echo experiments have been performed in an oil bath using a planar target 3 mm away from the array. The measurement results show a signal-to-noise ratio of 16.65 dB with 122 μW power consumption around 3M Hz. By using an open-loop configuration, the leakage current of a CMUT device has also been characterized [76].

A new programmable floating-gate bump circuit has been proposed and fabricated in a 0.5 μm CMOS process. The height, the center and the width of the circuit bell-shaped transfer characteristics can be programmed individually. A multivariate radial basis function with a diagonal matrix can be realized by cascading these bump cells. Based on this bump circuit, a novel compact RBF-based soft classifier and, with an addition of a simple current mode winner-take-all circuit, a 16×16 analog vector quantizer have been designed, fabricated, and tested. By using receiver operating characteristic curves as evaluation measures, the performance of the analog classifiers are comparable to digital counterparts. The

measured power efficiency is estimated as $513MMAC/s/mW$, which is at least two orders of magnitude better than digital signal processors. Automatic gender identification experiments are demonstrated using this analog vector quantizer with an accuracy around 70% [77, 78].

An analog VLSI approach to implementing a projection neural network for solving the optimization problem of a support vector machine has been proposed. The kernel function can be realized by a floating-gate bump circuit with a tunable width. The analog signal processing circuits make use of the intrinsic nonlinearity of the silicon devices without employing any resistors or operational amplifiers. Therefore, this approach is suitable for large scale analog VLSI implementations. The projection neural network for the support vector machine learning and classification has been verified on the transistor level using a SPICE simulator. With these analog signal processing techniques, a low-power adaptive analog system with the capability of learning, classifying, and regression becomes feasible. It can be integrated with the sensor interface circuits and results in a highly efficient smart sensor system without employing any analog-to-digital converter [79].

REFERENCES

[1] C. M. Twigg, *Floating Gate Based Large-Scale Field-Programmable Analog Array for Analog Signal Processing*. Atlanta, GA: Ph.D. Dissertation, Georgia Institute of Technology, 2006.

[2] V. Srinivasan, G. J. Serrano, J. Gray, and P. Hasler, "A precision cmos amplifier using floating-gate transistors for offset cancellation," *IEEE Journal of Solid–State Circuits*, vol. 42, no. 2, pp. 280–291, 2007.

[3] E. Ozalevli, H.-J. Lo, and P. Hasler, "Binary-weighted digital-to-analog converter design using floating-gate voltage references," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 55, no. 4, pp. 990–998, 2008.

[4] G. J. Serrano and P. Hasler, "A precision cmos amplifier using floating-gate transistors for offset cancellation," *IEEE Journal of Solid–State Circuits*, vol. 43, no. 2, pp. 558–565, 2008.

[5] T. S. Hall, C. M. Twigg, J. D. Gray, P. Hasler, and D. V. Anderson, "Large-scale field-programmable analog arrays for analog signal processing," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 52, no. 11, pp. 2298–2307, 2005.

[6] M. Kucic, A. Low, P. Hasler, and J. Neff, "A programmable continuous-time floating-gate fourier processor," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 48, no. 1, pp. 90–99, 2001.

[7] P. Hasler, B. A. Minch, and C. Diorio, "An autozeroing floating-gate amplifier," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 48, no. 1, pp. 74–82, 2001.

[8] P. E. Hasler, *Foundations of Learning in Analog VLSI*. Pasadena, CA: Ph.D. Dissertation, California Institute of Technology, 1997.

[9] A. Bandyopadhyay, G. Serrano, and P. Hasler, "Adaptive algorithm using hot-electron injection for programming analog computational memory elements within 0.2% of accuracy over 3.5 decades," *IEEE Journal of Solid–State Circuits*, vol. 41, no. 9, pp. 2107–2114, 2006.

[10] J. Ramirez-Angulo, A. J. Lopez-Martin, and R. G. C. andF. M. Chavero, "Very low-voltage analog signal processing based on quasi-floating gate transistors," *IEEE Journal of Solid–State Circuits*, vol. 39, no. 3, pp. 434–442, 2004.

[11] R. Schreier, J. Silva, J. Steensgaard, and G. C. Temes, "Design-oriented estimation of thermal noise in switched-capacitor circuits," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 52, no. 11, pp. 2358–2368, 2005.

[12] N. A. Hall, B. Bicen, M. K. Jeelani, W. Lee, S. Qureshi, F. L. Degertekin, and M. Okandan, "Micromachined microphones with diffraction-based optical displacement detection," *Journal of the Acoustical Society of America*, vol. 118, no. 5, pp. 3000–3009, 2005.

[13] P. Hasler, "Continuous-time feedback in floating-gate mos circuits," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 48, no. 1, pp. 56–64, 2001.

[14] B. Khuri-Yakub, C.-H. Cheng, F. Degertekin, S. Ergun, S. Hansen, X. Jin, and O. Oralkan, "Silicon micromachined ultrasonic transducers," *Japanese Journal of Applied Physics*, vol. 39, no. 5B, pp. 2883–2887, 2000.

[15] J. Knight, J. McLean, and F. Degertekin, "Low temperature fabrication of immersion capacitive micromachined ultrasonic transducers on silicon and dielectric substrates," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 51, no. 10, pp. 1324–1333, 2004.

[16] R. R. Harrison and C. Charles, "A low-power low-noise cmos amplifier for neural recording applications," *IEEE Journal of Solid–State Circuits*, vol. 38, no. 6, pp. 958–965, 2003.

[17] C. C. Enz and G. C. Temes, "Circuit techniques for reducing the effects of op-amp imperfections: autozeroing, correlated double sampling, and chopper stabilization," *IEEE Proceedings*, vol. 84, no. 11, pp. 1584–1614, 1996.

[18] W. Bracke, P. Merken, R. Puers, and C. V. Hoof, "Ultra-low-power interface chip for autonomous capacitive sensor systems," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 54, pp. 130–140, 2007.

[19] S. L. Garverick, M. L. Nagy, N. K. Rao, and A. P. D. K. Hartsfield, "A capacitive sensing integrated circuit for detection of micromotor critical angles," *IEEE Journal of Solid–State Circuits*, vol. 32, no. 1, pp. 23–30, 1997.

[20] J. Wu, G. K. Fedder, and L. R. Carley, "A low-noise low-offset capacitive sensing amplifier for a $50-\mu g/\sqrt{Hz}$ monolithic cmos mems accelerometer," *IEEE Journal of Solid–State Circuits*, vol. 39, no. 5, pp. 722–730, 2004.

[21] A. Sadat, C. Y. H. Qu, J. S. Yuan, and H. Xie, "Low-power cmos wireless mems motion sensor for physiological activity monitoring," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 52, no. 12, pp. 2539–2551, 2005.

[22] M. Tavakoli and R. Sarpeshkar, "An offset-canceling low-noise lock-in architecture for capacitive sensing," *IEEE Journal of Solid–State Circuits*, vol. 38, no. 2, pp. 244–253, 2003.

[23] M. Baker and R. Sarpeshkar, "A low-power high-psrr current-mode microphone preamplifier," *IEEE Journal of Solid–State Circuits*, vol. 38, no. 10, pp. 1671–1678, 2003.

[24] M. Pederson, W. Olthuis, and P. Bergveld, "High-performance condenser microphone with fully integrated cmos amplifier and dc-dc voltage converter," *IEEE/ASME Journal of Microelectromechanical Systems*, vol. 7, no. 4, pp. 387–394, 1998.

[25] C. Furst, "A low-noise/low-power preamplifier for capacitive microphones," in *IEEE Proceedings of the International Symposium on Circuits and Systems*, May 1996.

[26] J. Silva-Martinez and J. Alcedo-Suner, "A cmos preamplifier for electret microphones," in *IEEE Proceedings of the International Symposium on Circuits and Systems*, May 1995.

[27] F. L. Degertekin, R. O. Guldiken, and M. Karaman, "Annular-ring cmut arrays for forward-looking ivus: transducer characterization and imaging," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 53, no. 2, pp. 474–482, 2006.

[28] O. Oralkan, *Acoustical Imaging Using Capacitive Micromachined Ultrasonic Transducer Arrays: Devices, Circuits, and Systems*. Stanford, CA: Ph.D. Dissertation, Stanford University, 2004.

[29] R. O. Guldiken, J. Zahorian, M. Balantekin, F. L. Degertekin, C. Tekes, A. Sisman, and M. Karaman, "Dual-annular-ring cmut array for forward-looking ivus imaging," in *IEEE Proceedings of the Ultrasonics Symposium*, Oct. 2006.

[30] T. Delbruck, "Bump circuits for computing similarity and dissimilarity of analog voltage," in *IEEE Proceedings of the International Neural Network Society*, pp. 475–479, Oct. 1991.

[31] S. S. Watkins and P. M. Chau, "A radial basis function neurocomputer implemented with analog vlsi circuits," in *Proceedings of the International Joint Conference on Neural Networks*, July 1992.

[32] J. Choi, B. J. Sheu, and J. C.-F. Chang, "A gaussian synapse circuit for analog vlsi neural networks," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 2, no. 1, pp. 129–133, 1994.

[33] S.-Y. Lin, R.-J. Huang, and T.-D. Chiueh, "A tunable gaussian/square function computation circuit for analog neural networks," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 45, no. 3, pp. 441–446, 1998.

[34] D. S. Masmoudi, A. T. Dieng, and M. Masmoudi, "A subthreshold mode programmable implementation of the gaussian function for rbf neural networks applications," in *IEEE Proceedings of the International Symposium on Intelligent Control*, Oct. 2002.

[35] T. Yamasaki and T. Shibata, "Analog soft-pattern-matching classifier using floating-gate mos technology," *IEEE Transactions on Neural Networks*, vol. 14, no. 5, pp. 1257–1265, 2003.

[36] P. Hasler, P. Smith, C. Duffy, C. Gordon, J. Dugger, and D. Anderson, "A floating-gate vector-quantizer," in *IEEE Proceedings of the Midwest Symposium on Circuits and Systems*, Aug. 2002.

[37] D. Hsu, M. Figueroa, and C. Diorio, "A silicon primitive for competitive learning," vol. 13, pp. 713–719, 2001.

[38] B. A. Minch, "Evolution of a folded floating-gate differential pair," in *IEEE Proceedings of the Midwest Symposium on Circuits and Systems*, Aug. 2000.

[39] P. Hasler and J. Dugger, "An analog floating-gate node for supervised learning," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 52, no. 5, pp. 834–845, 2005.

[40] M.Ogawa, K. Ito, and T. Shibat, "A general-purpose vector-quantization processor employing two-dimensional bit-propagating winner-take-all," in *Proceedings of the Symposium on VLSI Circuits*, June 2002.

[41] M. Bracco, S. Ridella, and R. Zunino, "Digital implementation of hierarchical vector quantization," *IEEE Transactions on Neural Networks*, vol. 14, no. 5, pp. 1072–1084, 2003.

[42] G. Cauwenberghs and V. Pedron, "A low-power cmos analog vector quantizer," *IEEE Journal of Solid–State Circuits*, vol. 32, no. 8, pp. 1278–1283, 1997.

[43] J. Glossner, K. Chirca, M. Schulte, H. Wang, N. Nasimzada, D. Har, S. Wang, A. J. Hoane, G. Nacer, M. Moudgill, and M. S. Vassiliadis, "Sandblaster low power dsp," in *IEEE Proceedings of the Custom Integrated Circuits Conference*, Oct. 2004.

[44] H. G. Hirsh and D. Pearce, "The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *ISCA ITRW Automatic Speech Recognition*, Sept. 2000.

[45] P. Hasler, P. Smith, C. Duffy, C. Gordon, J. Dugger, and D. Anderson, "A floating-gate vector-quantizer," in *IEEE Proceedings of the Midwest Symposium on Circuits and Systems*, Aug. 2002.

[46] D. Hsu, M. Figueroa, and C. Diorio, "Competitive learning with floating-gate circuits," *IEEE Transactions on Neural Networks*, vol. 13, pp. 732–744, 2002.

[47] R. R. Harrison, J. A. Bragg, P. Hasler, B. A. Minch, and S. P. Deweerth, "A cmos programmable analog memory-cell array using floating-gate circuits," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 48, no. 1, pp. 4–11, 2001.

[48] Y.-S. Xia and J. Wang, "Global asymptotic and exponential stabiity of a dynamic neural system with asymmetric connection weights," *IEEE Transactions on Automatic Control*, vol. 46, pp. 635–638, 2001.

[49] Y.-S. Xia and J. Wang, "A one-layer recurrent neural network for support vector machine learning," *IEEE Transactions on Systems, Man, and Cybernetics XPart B: Cybernetics*, vol. 34, pp. 1261–1269, 2004.

[50] R. Genov and G. Cauwenberghs, "Analog neural network for support vector machine learning," *IEEE Transactions on Neural Networks*, vol. 17, pp. 1085–1091, 2006.

[51] V. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.

[52] R. J. Vanderbei, "Loqo:an interior point code for quadratic programming," *Optimization Methods and Software*, vol. 11, pp. 451–484, 1999.

[53] T. Joachims and J. Platt, *Advances in Kernel Methods–Support Vector Learning*. Cambridge, MA: The MIT Press, 1999.

[54] D. Anguita, A. Boni, and S. Ridella, "A digital architecture for support vector machines: Theory, algorithm and fpga implementation," *IEEE Transactions on Neural Networks*, vol. 14, pp. 993–1009, 2003.

[55] R. Genov and G. Cauwenberghs, "Kerneltron: Support vector "machine" in silicon," *IEEE Transactions on Neural Networks*, vol. 14, pp. 1426–1434, 2003.

[56] S. Chakrabartty and G. Cauwenberghs, "Sub-microwatt analog vlsi trainable pattern classifier," *IEEE Journal of Solid–State Circuits*, vol. 42, pp. 1169–1179, 2007.

[57] D. Tank and J. Hopfield, "Simple "neural" optimization networks: An a/d converter, signal decision circuit, and a linear programming circuit," *IEEE Transactions on Circuits and Systems*, vol. 33, pp. 533–541, 1986.

[58] M. P. Kennedy and L. O. Chua, "Neural networks for nonlinear programming," *IEEE Transactions on Circuits and Systems*, vol. 35, pp. 554–562, 1988.

[59] D. Anguita, S. Ridella, and S. Rovetta, "Circuital implementation of support vector machines," *IEEE Electronics Letters*, vol. 34, pp. 1596–1597, 1998.

[60] S. Zhang and A. G. Costantinides, "Lagrange programming neural networks," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 39, pp. 441–452, 1992.

[61] A. Bouzerdom and T. R. Pattison, "Neural network for quadratic optimization with bound constraints," *IEEE Transactions on Neural Networks*, vol. 4, pp. 293–304, 1993.

[62] K. Urahama, "Gradient projection network: Analog solver for linearly constrained nonlinear programming," *Neural Computation*, vol. 8, pp. 1061–1073, 1996.

[63] Y. Leung, K.-Z. Chen, and X.-B. Gao, "A high-performance feedback neural network for solving convex nonlinear programming problems," *IEEE Transactions on Neural Networks*, vol. 14, pp. 1469–1477, 2003.

[64] X.-Y. Wu, Y.-S. Xia, J. Li, and W.-K. Chen, "A high performance neural network for solving linear and quadratic programming problems," *IEEE Transactions on Neural Networks*, vol. 7, pp. 67–72, 1996.

[65] D. Anguita and A. Boni, "Towards analog and digital hardware for support vector machines," in *Proceedings of the International Joint Conference on Neural Networks*, July 2001.

[66] Y.-S. Xia and J. Wang, "A general methodology for designing globally convergent optimization neural networks," *IEEE Transactions on Neural Networks*, vol. 9, pp. 1331–1343, 1998.

[67] Y.-S. Xia, H. Leung, and J. Wang, "A projection neural network and its application to constrained optimization problems," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 4, pp. 447–458, 2002.

[68] Y. Tan, Y. Xia, and J. Wang, "Neural network realization of support vector methods for pattern classification," in *Proceedings of the International Joint Conference on Neural Networks*, July 2000.

[69] D. Anguita and A. Boni, "Improved neural network for svm learning," *IEEE Transactions on Neural Networks*, vol. 5, pp. 1243–1244, 2002.

[70] D. Anguita and A. Boni, "Neural network learning for analog vlsi implementations of support vector machines: a survey," *Neuralcomputing*, vol. 55, pp. 265–283, 2003.

[71] B. Scholkopf and A. J. Smola, *Learning with Kernels–Support Vector Machines, Regularization, Optimization and Beyond*. Cambridge, MA: The MIT Press, 2002.

[72] O. L. Mangasarian, *Nonlinear Programming*. New York: McGraw-Hill, 1969.

[73] M. E. Primak and A. Zeleke, "Set variational inequalities," *Applied Mathematics and Computation*, vol. 85, pp. 165–180, 1997.

[74] D. Kinderlehrer and G. Stampcchia, *An Introduction to Variational Inequalities and Their Applications*. New York: Society for Industrial Mathematics, 1980.

[75] S.-Y. Peng, M. S. Quershi, P. E. Hasler, A. Basu, and F. L. Degertekin, "A charge-based low-power high-snr capacitive sensing interface circuit," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*.

[76] S.-Y. Peng, M. S. Qureshi, A. Basu, R. O. Guldiken, F. L. Degertekin, and P. E. Hasler, "Floating-gate based cmut sensing circuit using capacitive feedback charge amplifier," in *IEEE Proceedings of the Ultrasonics Symposium*, Oct. 2006.

[77] S.-Y. Peng, P. E. Hasler, and D. V. Anderson, "An analog programmable multidimensional radial basis function based classifier," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 54, pp. 2148–2158, 2007.

[78] S.-Y. Peng, P. E. Hasler, and D. Anderson, "A programmable analog radial-basis-function based classifier," in *INNS Proceedings of the International Conference on Acoustics, Speech, and Signal*, March 2008.

[79] S.-Y. Peng, B. A. Minch, and P. E. Hasler, "Analog vlsi implementation of support vector machine learning and classification," in *IEEE Proceedings of the International Symposium on Circuits and Systems*, May 2008.