POWER-EFFICIENT ANALOG SYSTEMS TO PERFORM SIGNAL-PROCESSING USING FLOATING-GATE MOS DEVICE FOR PORTABLE APPLICATIONS

A Dissertation
Presented to
The Academic Faculty

By

Ravi Chawla

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

School of Electrical and Computer Engineering
Georgia Institute of Technology
December 2004

POWER-EFFICIENT ANALOG SYSTEMS TO PERFORM SIGNAL-PROCESSING USING FLOATING-GATE MOS DEVICE FOR PORTABLE APPLICATIONS

Approved by:

Dr. Paul Hasler, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. David Anderson
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Joy Laskar
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Mark T. Smith
Hewlett-Packard Laboratories
Palo Alto, California

Dr. Phil Allen
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: December 2004

DEDICATION

To my parents and my brother

ACKNOWLEDGEMENTS

I wish to gratefully acknowledge my advisor, Dr. Paul Hasler, for helping me during my stay at Gatech, providing an opportunity, guiding my research and reviewing this thesis. I also want to thank Dr. Joy Laskar, Dr. Phil Allen, Dr. David Anderson, and Dr. Mark Smith for all the discussions we had and for reviewing this thesis. Many thanks also to the committee for the fruitful reviews. I would like to thank Guillermo Serrano for the wonderful office mate he has been during my stay at ICE-LAB. I had a very enjoyable time working with all the members of icelab specially Abhishek Bandyopadhyay, Venkatesh Srinivasan, Chris Twigg, Ryan Robucci and Dave Graham.

I would like to thank the most important people in my life: my parents and my brother. Without their love, support and belief, I would not be able to become anywhere close to what I am today. So, thank you for everything.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Digital Signal Processors (DSPs) have been an important component of all signal processing systems for over two decades now. Some of the obvious advantages of digital signal processing are the flexibility to make specific changes in the processing functions through hardware or software programming, faster processing speeds of the DSPs, cheaper storage, and retrieval of digital information and lower sensitivity to electrical noise.

The explosive growth of wireless and signal processing applications has resulted in an increasing demand for such systems with low cost, low power consumption, and small form factors. With high–level of integration to single–chip systems, power consumption becomes a very important concern to be addressed. Intermediate–Frequency (IF) band signal processing requires the use of an array of DSPs, operating in parallel, to meet the speed requirements [1]. This is a power intensive approach and makes use of certain communication schemes impractical in applications where power budget is limited. The front–end ADC and back–end DAC converters required in these systems become expensive when the signal is of wideband nature and a greater resolution is required.

We present techniques to use floating–gate devices to implement signal processing systems in the analog domain in a power efficient and cost effective manner. Use of floating–gate devices mitigates key limitations in analog signal processing such as the lack of flexibility to specific changes in processing functions and the lack of programmability. This will impact the way a variety of signal processing systems are designed currently. It also enables array signal processing to be done in an area efficient manner. As will be shown through sample applications, this methodology promises to replace expensive wideband ADC and DAC converters with relatively easy to implement baseband data converters and an array of power intensive high

speed DSPs with baseband DSPs. This approach is especially beneficial for portable systems where a lot of applications are running from a single battery.

CHAPTER 1

OVERVIEW

1.1 DSP versus ASP

Currently, almost all signal processing techniques convert the incoming analog signal to digital domain after some basic analog blocks such as LNA, mixer and filter. Thereafter, the computation/processing on the signal is done using DSPs. The popularity of this scheme is due to the limitations of analog signal processing (ASP) such as [2]

- Accuracy limitations due to tolerances.
- Limited repeatability of response.
- Lack of flexibility to specific changes in processing functions.
- Sensitivity to electrical noise.

In contrast, digital signal processing was able to overcome all these limitations but had disadvantages of its own. Using DSPs is not always the power efficient way to build systems. In today's world, where the trend is to have portable solutions with good graphics, data and audio quality while maximizing the battery life, a DSP microprocessor using watts of power may not be the ideal solution. This problem becomes even more critical in cases of array signal processing, where an array of DSPs are required. This may be hard to implement in a power efficient way with a high level of integration.

1.2 Cooperative Analog/Digital Signal Processing (CADSP)

With recent focus on integrating large high–performance, low–cost systems, neither analog signal processing nor digital signal processing can exist by itself. While the

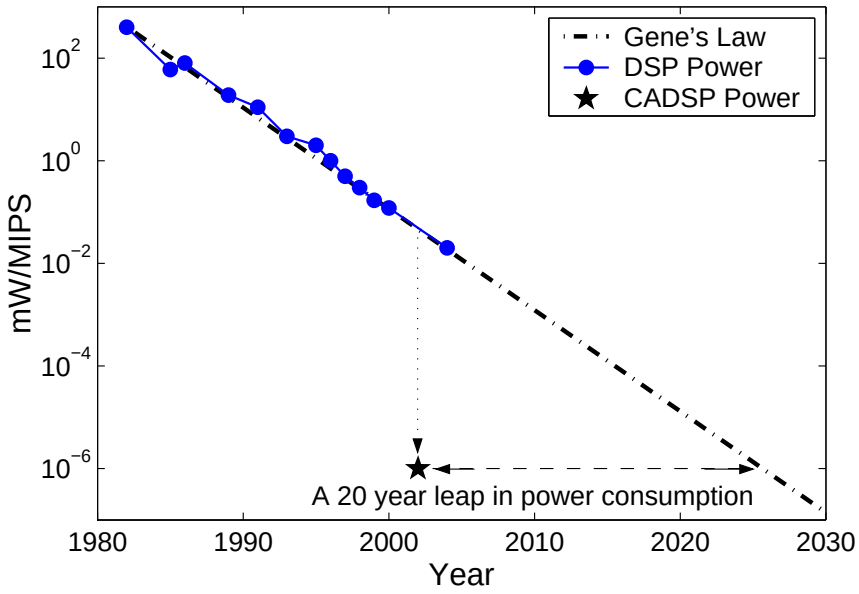


Figure 1. Gene's law showing computation versus power consumption.: Gene's law showing the power consumption for performing computation using DSPs as the technology improves with years. The plot also shows the power consumed when a programmable analog system is used to perform similar kind of computation. It is clearly evident that using analog systems where possible along with digital helps in reducing the power consumption by orders of magnitude. [3]

real world interface is purely analog, most of the modern communication systems are digital in nature. Among the limitations discussed above, lack of flexibility or programmability after fabrication is the major limitation that keeps analog systems from being used for a variety of signal processing systems. Typically, analog circuits are only limited to front–end processing. However, designing programmable analog systems opens up a whole new way of designing signal processing systems. These analog VLSI systems can be compact and give comparable performance while being extremely low power at the same time. Figure 1 shows the plot popularly known as Gene's Law showing the power consumed for performing computation [3]. The plot also shows the power consumed when a programmable analog system is used to perform similar kind of computation. It is clearly evident that using analog systems where possible along with digital helps in reducing the power consumption by orders of magnitude.

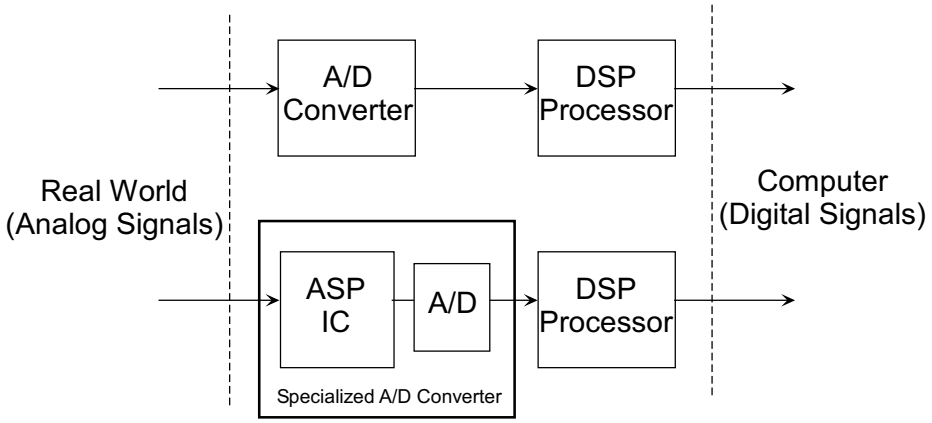


Figure 2. Cooperative analog/digital signal processing (CADSP) approach.: Traditional approach involve converting the incoming analog signal to digital as soon as possible and then perform all the computation digitally. An alternate approach is to perform some of the computation using analog signal processing and then convert them into digital. This is called the CADSP approach. This approach leads to simpler, easy to design A/D converters and reduces the computation complexity of DSPs. CADSP approach leads to a more power–efficient design. The exact boundary between analog and digital depends upon the particular application.

We define Cooperative Analog/Digital Signal Processing (CADSP) approach as processing signals as much as possible in the analog domain before converting them into digital in order to design low–power systems. This approach enables architectures or implementations where both programmable ASP systems and DSP systems are used together to perform signal processing for real–world signals. Figure 2 shows a simple block diagram of how CADSP approach can be implemented in current systems. The goal is to enhance the total performance of the system by utilizing both analog and digital together in a mutually beneficial way. This can only be achieved by maintaining a balance between the two approaches for optimal performance without over-doing something. The right partition, as shown in Fig. 2, between the analog and digital signal processing blocks is a problem that has to be solved on a case by case basis.

1.3 Motivation for presented research

Using DSPs to perform IF-band or all the baseband processing can be extremely power consuming and impractical especially for portable applications [1]. Thus, this approach limits the use of certain communication schemes and makes them impractical due to their power requirements. Also, the front–end ADC and back–end DAC converters required in these systems become expensive when the signal is of wideband nature and a greater resolution is required.

In this thesis, we investigate how to design programmable analog systems to perform signal processing that would be usually done using DSPs. We present design of analog signal processing systems using floating–gate devices designed to operate at the desired frequency in a power efficient way. This will enable the audio band signal processing techniques, using floating–gate devices, to become useful at IF band. We attempt to answer questions regarding performance such as speed and signal-to-noise ratio (SNR) of floating–gate devices that will enable them to become a part of the main stream analog circuit design. The programmability feature of floating–gate devices can be exploited in many ways: tuning of circuits after fabrication to get desired responses (in programmable filters, multipliers), removal of offsets (in differential pairs), providing bias currents (also tunable to get desired performance) and improvement in linearity due to capacitive attenuation, if desired [4, 5, 6, 7, 8, 9, 10, 11]. The floating–gate array architecture also enables to perform computation in a parallel fashion further improving speed of the system. This also enables array signal processing with compact and power–efficient architectures.

The thesis is organized into eight chapters. In Chapter 2, we briefly review floating–gate device and compare them with standard MOS device for performance such as frequency response and SNR. We will also discuss the algorithm that is used to program the floating–gate devices accurately in an array. We conclude the chapter with a discussion on power–efficient design and how to maximize bandwidth for a

given power. In Chapter 3, we propose a fully programmable floating–gate analog vector–matrix multiplier that can be used for a variety of signal processing applications. We discuss the governing design equations for the core multiplier cell along with measured experimental results. Chapter 4 presents the architecture and implementation for a programmable analog modulator/demodulator (PAMD) using floating–gate devices. We present the various blocks along with the measured results from a test chip showing the potential applications. PAMD can be used for any communication system requiring arbitrary waveform generation. In Chapter 5, we discuss the design of traditional programmable OTA-C based continuous time filters that are needed for signal processing applications. We describe the design of programmable Operational Transconductance Amplifiers (OTAs) along with their measured results. Chapter 6 presents a compact and power efficient programmable bandpass filter section based on the current–conveyor structure. We also present the results from a $10th$-order filter composed of the programmable second–order sections that can be programmed to any filter transfer function such as Butterworth, Chebyshev. In Chapter 7, we present the design of highly–linear current–mode log–domain filter sections using floating–gate devices that can be used in anywhere up till few MHz. We conclude the thesis in Chapter 8 with the impact of the presented work along with some future directions.

CHAPTER 2

FLOATING–GATE OVERVIEW AND ANALYSIS

As briefly mentioned earlier, one of the major limitations of analog signal processing (ASP) systems was lack of flexibility or ease of programmability as compared to digital signal processors (DSPs). In this thesis, we use our floating–gate (FG) MOS devices to overcome this limitation of ASP systems. We will be using FG devices for designing programmable analog systems for a variety of signal processing applications. In this chapter, we will discuss the schematic and layout of a typical FG device along with the basic DC transfer equations. We will also discuss how to perform accurate programming of an array of these elements. We will also compare the performance such as frequency response and signal–to–noise ratio (SNR) of a standard MOS transistor with our FG transistor.

2.1 Floating–Gate Device

Figure 3 shows the layout, cross–section and circuit symbol for a floating–gate pFET. A floating–gate is a MOS gate surrounded by silicon–dioxide with no DC path to ground and hence, the name floating–gate. Charge on the floating–gate is stored permanently, providing a long–term memory, because it is completely surrounded by a high–quality insulator. This device is very similar to the one used in digital EEPROM memories. Floating–gate devices can be used as circuit elements for a variety of analog circuit applications [6, 7, 8, 9, 10, 11].

From the layout in Fig. 3, we see that the floating–gate is the gate of a MOSFET with no contacts to other layers. This gate can be capacitively coupled to other layers. The floating–gate voltage, determined by the charge stored on the floating gate, can modulate a channel between a source and drain, and therefore, can be used in computation. Floating–gate circuits provide IC designers with a practical,

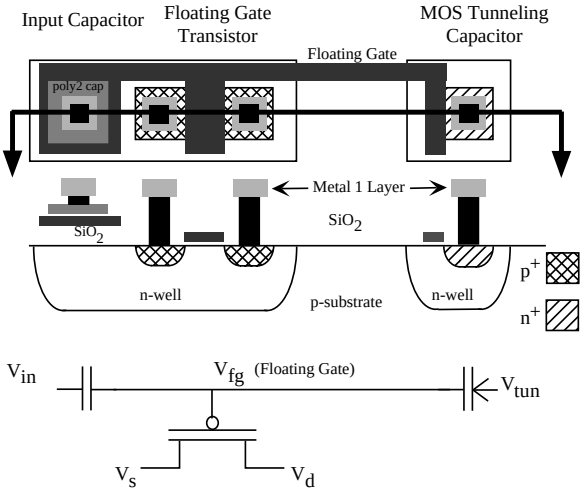


Figure 3. Layout, cross section and circuit diagram of the floating-gate pFET in a standard double–poly, n-well MOSIS process: The cross section corresponds to the horizontal line slicing through the layout view.The pFET transistor is the standard pFET transistor in the n-well process. The gate input capacitively couples to the floating–gate by either a poly–poly capacitor, a diffused linear capacitor, or a MOS capacitor, as seen in the circuit diagram (not explicitly shown in the other two figures). Between V_{tun} and the floating–gate is our symbol for a tunneling junction, a capacitor with an added arrow designating the charge flow.

capacitor–based technology since capacitors, rather than resistors, are a natural result of a MOS process. Floating–gate devices can compute a wide range of static and dynamic translinear functions by the particular choice of capacitive couplings into floating–gate devices [12, 13, 14, 15]. The charge on a floating–gate device can be modified using any of the following methods:

- **UV photo injection:** The charge on the floating–gate can be modified by using short–wave ultra violet (UV) light. Exposing silicon dioxide to UV light will impart enough energy to some carriers to overcome the oxide barrier. This method has been extensively used in the case of memory elements and can be used for normalizing arrays when there is huge disparity in the charge. This method although seemingly simple has drawbacks such as, lengthy programming time and difficulty in selectively programming elements in an array.

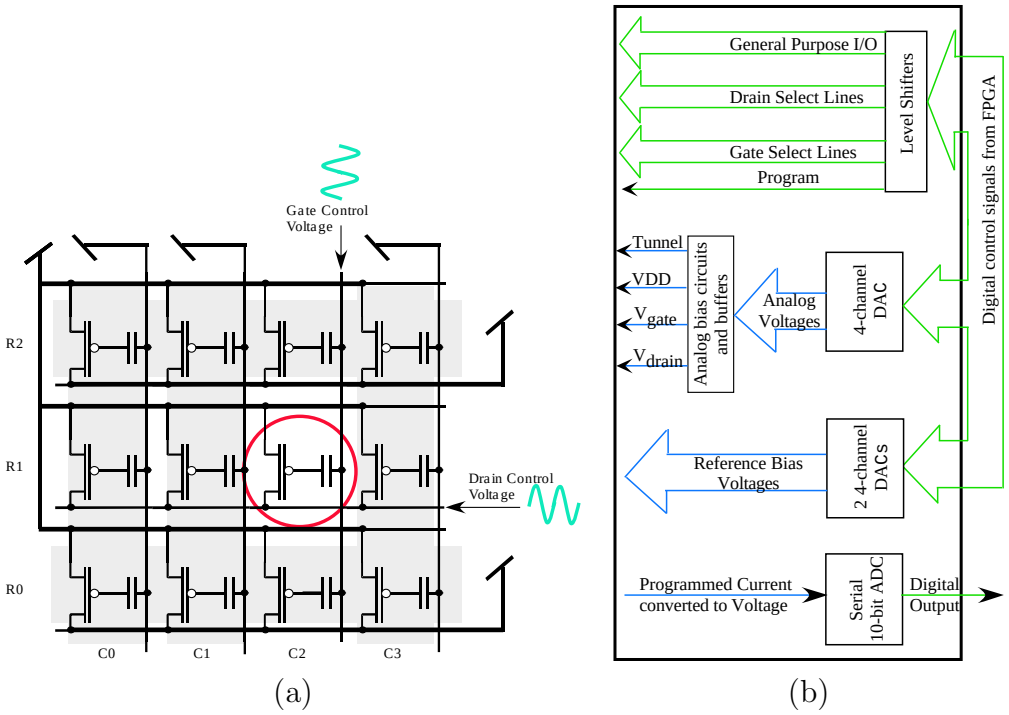


Figure 4. Computation and programming in floating–gate analog computing arrays: (a) Floating-gate array demonstrating element isolation by controlling the gate and drain voltage of each column and row. Selection of gate and drain voltages are controlled by on-chip multiplexor circuitry. (b) Block diagram of our custom programming board for automatic programming of large floating-gate arrays.

- **Fowler–Nordheim Electron Tunneling:** Tunneling is used to remove electrons from the floating–gate poly-silicon [16]. The tunneling junction is represented as a capacitor that couples the tunneling voltage terminal voltage to the floating–gate, as shown in Fig. 3. The arrow on the capacitor denotes the charge flow. Increasing the voltage across the n-well MOSCAP increases the effective electric voltage across the gate oxide, thereby increasing the probability of an electron escaping through the barrier. The voltage required for tunneling depends almost entirely on the oxide thickness. Since tunneling is an exponential of both field and the silicon–dioxide thickness, most of the current flows through where the oxide is thin. These spots are called "hot–spots". These currents can be really high and can lead to breaking of the lattice leaving open traps for free carriers. The amount of tunneling for floating–gate devices is based on the

voltage across the tunneling capacitor. In a 0.5μm process, a typical voltage of about 15V is required to get substantial tunneling.

- **Hot–Electron Injection:** Unlike tunneling, hot–electron injection is used to add electrons to the floating–gate node. Hot–electron injection is based on impact ionization. To have injection, two conditions must be met, a high current flowing through the transistor and a high gate to drain electric field. The impact ionization due to high energy holes travelling into the drain create excess electrons at the drain edge. These electrons travel back into the channel region and if their kinetic energy exceeds the silicon dioxide barrier, they can move across the oxide to the floating–gate poly–silicon. The impact ionization current is proportional to the pFET current and is the exponential function of the drain-to-channel potential (ϕ_{dc}). The rate of injection is decided by the drain-to-source voltage V_{ds} and the pulse width used.

The physical effects of hot–electron injection and electron tunneling become more pronounced as the line widths of existing processes are scaled down further[18], improving our floating–gate circuits. We use Fowler–Nordheim tunneling for a global erase in our floating–gate arrays and hot–electron injection for accurate programming of each element in the array.. The sub-threshold drain current of a floating-gate transistor in saturation, is given by,

$$I_d = I_o exp(\frac{-\kappa V_{fg} + V_s}{U_T}) exp(-\frac{V_d}{V_A}) \tag{1}$$

where $\kappa = \frac{C}{C_T}$ is the fractional change in the pFET's surface potential due to a ΔV_{fg} change in the floating-gate voltage, U_T is the thermal voltage expressed as KT/q, V_A is the early voltage, V_{fg} is the floating-gate voltage given by $V_{fg} = \frac{C}{C_T} \times V_g + V_{charge}$, V_{charge} is the charge stored on the floating-gate and C_T is the total gate capacitance. For a floating gate transistor operating in the above threshold regime, a change in the floating gate charge can be viewed as a change in the threshold voltage of the

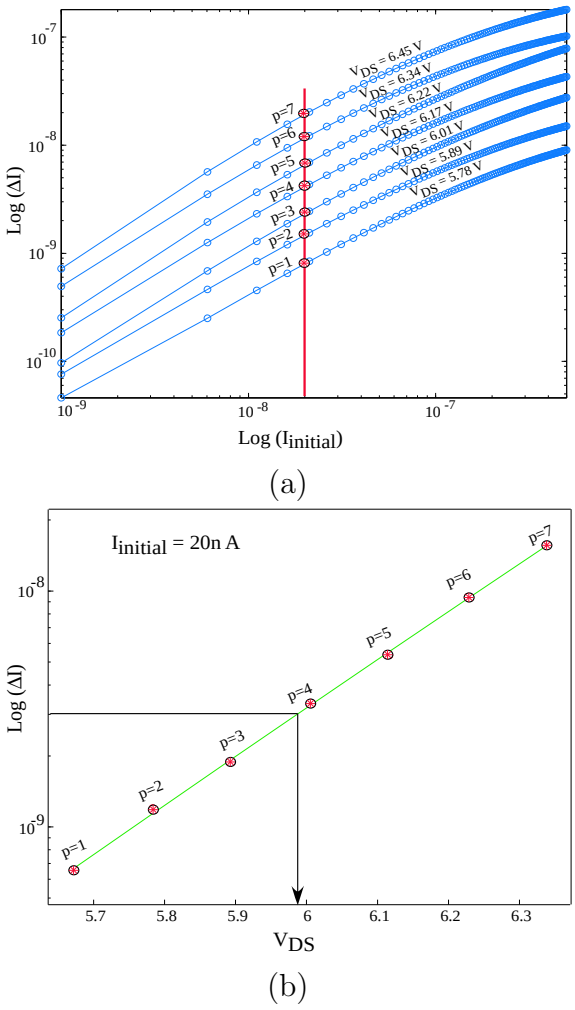


Figure 5. Characterization Curves for V_{ds} calibration. [17]: (a) Plot showing variation of injected currents for different initial currents as a function of different V_{ds}. (b) Sample plot showing change in current for different V_{ds} for initial current of 20nA. This plot is obtained from plot (a).

transistor. The drain current of a transistor in saturation is given by,

$$I_d = \frac{\kappa K}{2}(V_{sfg} - V_{th} + V_{charge})^2 \tag{2}$$

where V_{sfg} is V_s- V_{fg}, K is given by $\mu_p C_{ox}\frac{W}{L}$ and V_{th} is the threshold voltage of MOS device. The capacitor C should be chosen such that κ is as close to unity as possible. The value of κ determines the transconductance and gain of the transistor. For typical designs, C is chosen to be three times the gate capacitance C_{gate} to have κ

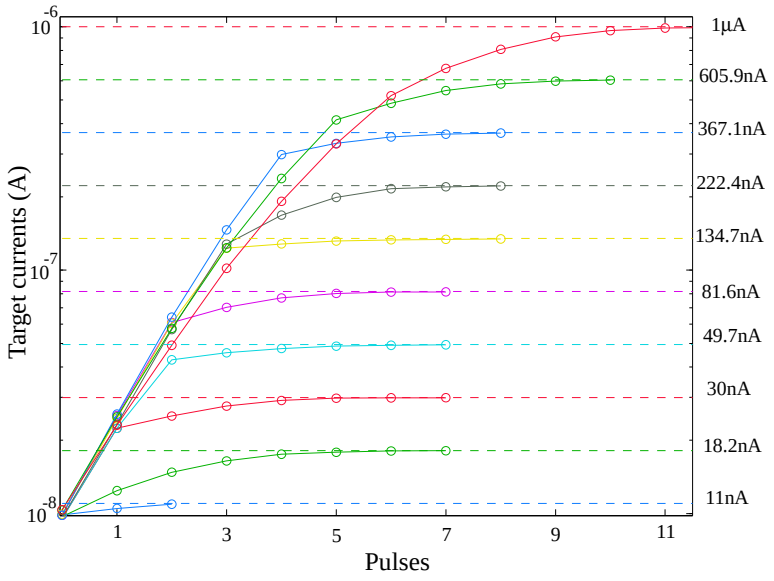


Figure 6. Programming accuracy and number of pulses [17].: Measurement showing asymptotic approach towards different target currents. The dash lines are the target currents. The average number of pulses required to hit a target current is 10-15 pulses.

of approximately 0.75. To enable use of large floating–gate arrays for building analog signal processing applications, accurately programming of these device elements individually become an important task. Programming large arrays of floating–gate elements requires systematic and automated methods. We use our adaptive programming algorithm to accurately program an array of these elements.

2.2 Floating–Gate Programming

One of the critical aspects in the design of a programmable analog system is the programming accuracy. There have been various implementations where floating–gates have been used. Floating-gates used in [19] employed a programming scheme similar to that used for EEPROMs based on electron tunneling [19]. This method requires a special oxide and at least a dual gate implementation adding extra fabrication steps. It also requires an extra switch per element to select the cell to be programmed, along with decoders, thereby increasing area per cell [19]. Also, this scheme uses small pulses of constant drain-to-source voltages (V_{ds}) that limits the programming

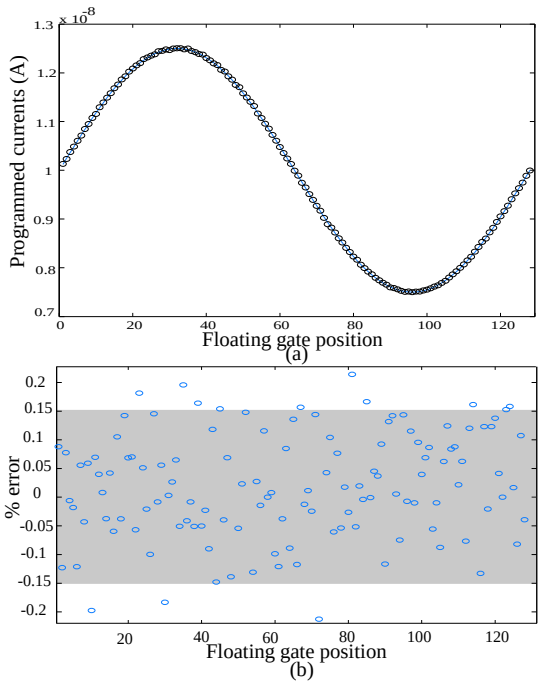


Figure 7. Measurement showing programming of floating–gates.: (a) A sine wave with 5 nA p–p and a DC of 10 nA was programmed onto 128 floating–gate elements; (b) Percentage error per element is shown.

accuracy to that obtained by a single pulse. The total program time increases with precision because of the logarithmic behavior of electron tunneling mechanism.

The programming scheme adopted for our floating–gate devices is based on both hot-electron injection and electron tunneling. Our method does not require any special oxide or extra gates to program floating-gates thereby enabling easy integration in a standard CMOS process. Figure 4 (a) shows that it is possible to isolate individual devices in a large matrix using peripheral control circuitry. We designed a custom programming board to program large floating-gate arrays. The board, shown in Fig. 4 (b), allows for flexible floating-gate array programming over a wide range of IC processes. The board interfaces with an FPGA that is controlled using a computer through an ethernet connection. The whole setup enables one to perform fast and accurate programming of floating–gate devices [20]. We will now discuss the programming algorithm along with error measurement for programming waveforms.

2.2.1 Programming Algorithm and Calibration

Our adaptive programming method enables us to perform accurate and fast programming [17]. The programming algorithm is a two step process. Floating–gate arrays are calibrated and coefficients for the particular chip are extracted using curve–fit. These coefficients are stored and are then used for accurate programming of the complete FG array. The algorithm computes the V_{ds} steps depending on the device current and the target current. This value is adjusted automatically as the device current approaches the target current. We will now describe our programming procedure [17].

2.2.1.1 Calibration Procedure

The calibration procedure to compute injection rate for different values of V_{ds} goes as follows:

1. Choose an element and pick a V_{ds} pulse voltage. The gate voltage is set such that the element has about 0.5nA of injection when pulsed with the chosen V_{ds}. Choose a value of t_{pulse}. This value will be a constant for entire programming algorithm.

2. Ramp up the whole array. While ramping up, all the voltages including drain, source, gate and tunnel are increased in small steps together. This is done to avoid having a large difference between any two nodes at any time.

3. Inject the element once by pulsing using chosen V_{ds} and constant t_{pulse}. Store the current value after the injection.

4. Ramp down the chip.

5. Repeat steps 1 to 4 for the same V_{ds} until the measured current exceed a threshold set for the calibration.

6. Repeat steps 1 to 5 for different values of V_{ds}s.

The change in current after each step is plotted versus the current before the pulse for different V_{ds}. This is shown in Fig. 5(a). A second–order curve fit is done to get better estimate for both sub–threshold and above–threshold current levels. The variation of log(δI) with V_{ds} has been plotted in Fig. 5(b). This was obtained using Fig. 5(a) and can be modelled as linear function.

2.2.1.2 Programming Procedure

The programming procedure to hit a target current is as follows:

1. Select the element to be injected. Connect the drain lines and gate lines of every other element to V_{dd}.

2. Measure the initial current. Use the initial current and the target current to compute the optimal V_{ds} required using the calibration data. Figure 5(b) is used to compute log(δI) versus V_{ds} for different value of initial current. An optimal V_{ds} can be computed from this (as shown in Fig 6) to achieve the target current.

3. If the computed V_{ds} is more than the ramped V_{dd} (6.5V), then the value of V_{dd} is used for V_{ds}.

4. Ramp up the chip and pulse using the computed V_{ds}. Only the selected element has the conditions necessary for injection.

5. Ramp down the chip. Measure the drain current. This value now becomes the new initial current.

6. Repeat steps 2 to 5 until the measured current equals the desired current.

The algorithm predicts the required V_{ds} for each element at each stage of injection. Typical number of steps required to hit a target are on an average 10-15 pulses are on Fig. 6. Fig. 7(a) shows sine-wave coefficients programmed on 128 floating–gate elements of a single row. The percentage error between the programmed value and

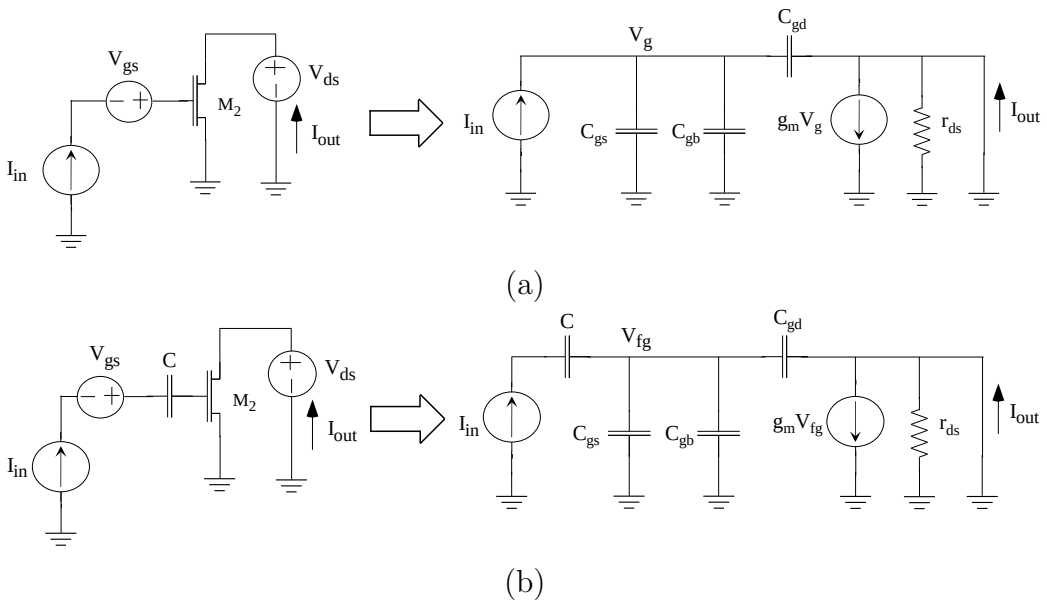


Figure 8. Small-signal model for computing the intrinsic transition frequency: (a) Circuit schematic and small-signal model to compute f_t of a MOS device. (b) Circuit schematic and small-signal model to compute f_t of a floating–gate device.

the target current is shown in Fig. 7(b). A worst case deviation of 0.2% is obtained with our programming scheme.

2.3 Floating–Gate Analysis

As can be seen from the previous section, floating–gate seems to be a promising technology for a wide variety of applications. However, there are still a few questions that need to be answered to help that process. How fast do floating–gate devices operate? Is the speed of floating–gate circuits comparable to the corresponding non-floating–gate circuits? What happens to the performance of circuits (like noise, linearity or dynamic range, SNR) when floating–gate devices are used? Theoretical analysis, as will be shown, suggests that there should not be much difference. If at all, floating-gate circuits give the added advantage of removing extra overheads that may be needed in traditional circuits for tunability, linearity improvements and offset removal.

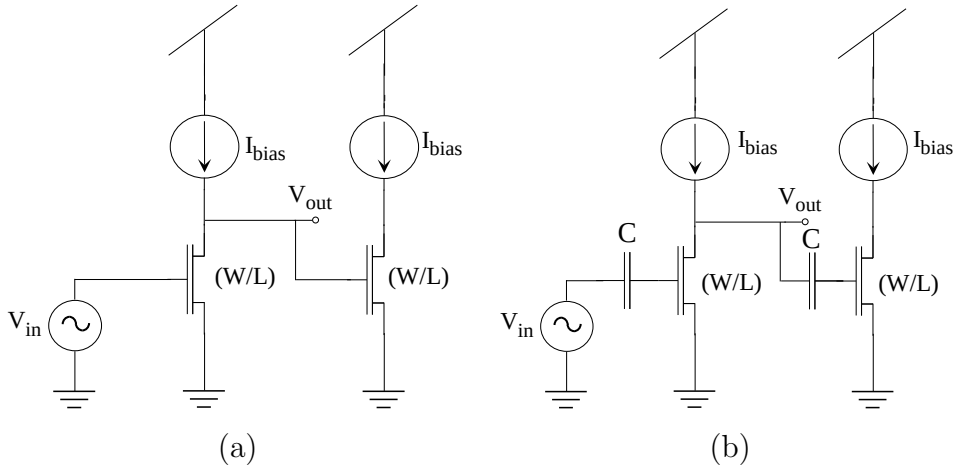


Figure 9. Circuit schematic for computing unity-gain frequency: (a) Circuit schematic of a MOS device driving another similar device used to compute the unity-gain frequency. (b) Circuit schematic of a floating–gate MOS device driving another similar device used to compute the unity-gain frequency.

2.3.1 Transition frequency of floating–gate devices

More often than not, the speed of analog circuits is defined by the cutoff frequency or the unity-gain frequency of the particular circuit, calculated from the small-signal model. This performance metric can be related to something called the *intrinsic transition frequency* of a MOS or BJT device. The *intrinsic transition frequency* or the *intrinsic cutoff frequency*, denoted as f_T, of a device is defined as the value of the frequency at which the short-circuit current gain of the device drops to unity. Simple analysis of the small-signal model of a MOS device, shown in Fig. 8 (a), shows that

$$f_T = \frac{g_m}{2\pi\left(C_{gs} + C_{gd} + C_{gb}\right)} \tag{3}$$

where g_m is the above-threshold or sub-threshold transconductance of the transistor, C_{gs}, C_{gd}, C_{gb} are intrinsic gate-source, gate-drain and gate-bulk capacitances of MOS device [21]. The f_T of a floating–gate device, shown in Fig 8 (b), can be defined using similar definition and is obtained to be

$$f_T = \frac{\kappa g_m}{2\pi\left(C_{gs} + C_{gd} + C_{gb}\right)} \tag{4}$$

where κ is defined as $\frac{C}{C+C_{gate}}$ and C, C_{gate} are input capacitance and total gate capacitance of the floating–gate device, respectively.

Equation 3 and 4 for f_T for a nominal MOS device and a floating–gate device show that the two are indeed different. The f_T of the floating–gate device is a factor of κ less than that of a normal MOS device. Although this parameter is right to express the performance of a stand alone device, it's not essentially a right measure when these devices are used in a circuit.

Figure 9 shows two devices, nominal MOS and floating–gate, driving similar loads as would be the case in a practical circuit. Using the small-signal circuit equivalents for the two circuits shown in Fig. 9, the *unity-gain frequency*, f_o of the two circuits is found to be the same as

$$f_o = \frac{g_m}{2\pi\left(C_{gs} + C_{gd} + C_{gb}\right)} \tag{5}$$

Thus, the performance of the two devices is identical as long as they drive similar loads. In the case where the load is not same as the device driving it, cutoff frequency of the floating–gate device is a factor of κ less than that of a nominal MOS device.

2.3.2 Signal–to–Noise ratio of floating–gate devices

Another important performance parameter when designing signal processing systems is signal–to–noise ratio (SNR). The input signal swing for a floating–gate MOS device can be obtained as,

$$(V_{pp})_{FG} = \frac{1}{\kappa}(V_{pp})_{MOS} \tag{6}$$

The equivalent noise of a MOS transistor can be expressed as a voltage–source in series with the gate when the effect of the input impedance can be neglected. Considering the effect of thermal noise and flicker noise for fairly long channel devices, the noise source is given as,

$$v_n^2 = [4KT\frac{2}{3g_m}\delta f + \frac{K_f I_D \delta f}{f C_{ox} L^2 {g_m}^2}] \tag{7}$$

To obtain the equivalent noise for a floating–gate, reflect the noise of a nominal MOS device back to the input of the floating–gate. The resulting noise source would be given as,

$$v_n^2 = \frac{1}{\kappa^2}[4KT\frac{2}{3g_m}\delta f + \frac{K_f I_D \delta f}{f C_{ox} L^2 {g_m}^2}] \tag{8}$$

This leads to the simple relation between the noise power of a nominal MOS device and a floating–gate MOS device as,

$$(NoisePower)_{FG} = \frac{1}{\kappa^2}(NoisePower)_{MOS} \tag{9}$$

Using 6 and 9, the SNR can be computed to be

$$(SNR)_{FG} = (SNR)_{MOS} \tag{10}$$

Hence, for comparable device sizes the SNR of a floating–gate device is similar to the SNR of a nominal MOS device. This leads to the conclusion that performance of the FG device is comparable to that of a MOS device for designing analog systems. The FG devices can be used in a number of applications due to the flexibility they provide in changing their characteristics after fabrication.

2.4 Floating–Gate Applications

The programmability feature of floating–gate devices can be exploited in many ways: tuning of circuits after fabrication to get desired responses (in programmable filters, multipliers), removal of offsets (in differential pairs), providing bias currents (also tunable to get desired performance) and improve linearity due to capacitive attenuation, if desired. Floating-gate devices help in taking care of some of the fundamental limitations like matching, offsets, bias generation, which can be really challenging in array processing, after the circuits have been fabricated [4, 5, 22, 7, 8, 9, 10, 23].

In a MOS transistor, there are two dominant sources of error i.e. device dimension mismatch and threshold mismatch that causes mismatch between the two transistors.

The device mismatch is due to any random variation is the device edges that can cause the effective (W/L) to be different. The threshold voltage (V_{th}) mismatch is due to the variation in the doping profile in the channel region causing the V_{th}s to be different. These two effects can be statistically modelled and can be reduced by increasing the area [24]. However, increasing area leads to other trade-offs such as parasitic capacitances. There have been a lot of techniques presented to remove these offsets after fabrication. Most of them lead to storing the input–referred offset voltage due to these effects on a capacitor and then subtract it out in the normal operation. These techniques, although effective, require extra circuitry or switches and require the process to be repeated to refresh the charge on the capacitors. In case of floating–gate devices, any offset due to mismatches can be easily removed by changing the charge V_{charge} of the floating–gate [5]. This technique does not involve measuring the absolute offset voltage making it easier to correct. Procedure for correcting mismatch between two devices consists of programming the two devices to have same drain currents for identical node voltages. This leads to similar I-V curves for the two devices and hence, removes the offset. We will be using this property of FG devices in our circuits to correct for any offsets.

2.5 Power Efficient Design

The term "Low Power" can be misleading when doing analog design for a range of frequencies. Power consumption is a varying specification depending on the performance of the circuit. Milli-watts of power consumption can be a lot for an audio–band application while the same number can be low power for RF application such as CDMA transceiver circuits. Any circuit or system is low power or power–efficient system as long as it gives maximum performance (such as bandwidth or speed) for certain amount of power consumption. In this chapter, we will discuss the design of power–efficient systems for maximum bandwidth.

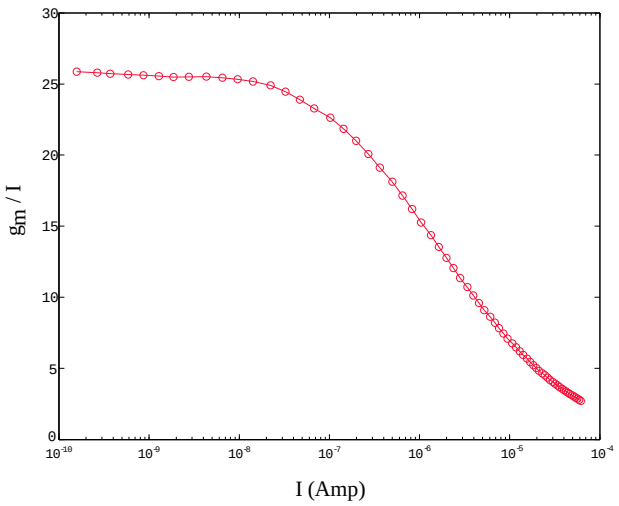


Figure 10. Plot showing g_m/I versus bias current, I.: Experimental result showing the variation of g_m/I versus bias current, I. It can be clearly seen from the plot that sub–threshold operation gives the most g_m, and hence the most bandwidth, for a particular bias current.

2.5.1 Maximum Frequency of Operation

Traditionally, the design of MOS IC's has always been done in the above-threshold region. With the increasing trend towards scaling down technologies, resulting in lower power supply voltages, designing in above–threshold region becomes non–trivial. The same circuits, however, can be designed with ease in sub–threshold region as less headroom is required in sub–threshold to keep devices in the saturation region of operation. In addition, the sub–threshold current-voltage relationship suggests that sub–threshold operation gives the maximum transconductance, g_m, for a particular bias current, as g_m is proportional to bias current, I. Thus to obtain most bandwidth for the amount of power consumed, circuits should be designed with transistors operating in sub–threshold (see Fig. 10).

As shown in 5, to get higher cutoff frequency, f_o, for the circuits, the transconductance and, hence, the current required to attain that f_o should be increased. But, to ensure sub–threshold operation, transistor size, $(\frac{W}{L})$, has to increase also. This in turn increases the inherent parasitic gate capacitance of the transistor and, thus,

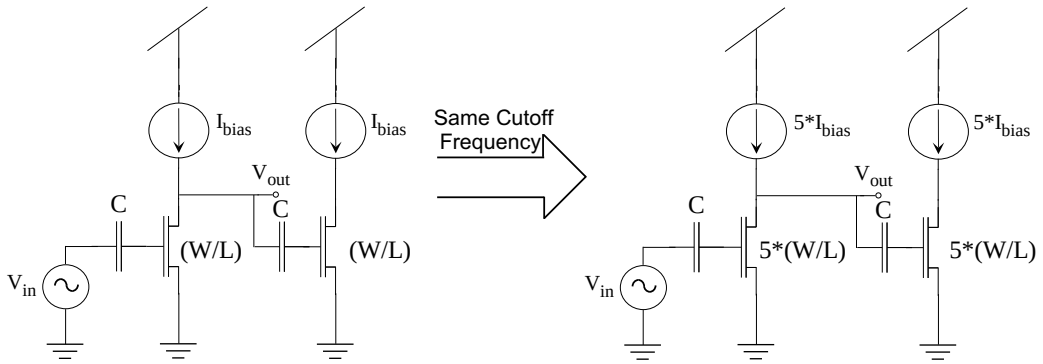


Figure 11. Effect of (W/L) on unity-gain frequency: Unity-gain frequency remains unchanged even if current and hence, (W/L) of the device is increased for sub-threshold operation. This is true if the load being driven is similar to the device driving it.

keeping f_o of the device same. Figure 11 illustrates this effect when driving similar loads. Thus, as long as the dominant capacitance in the circuit is the transistor capacitance, there will not be any gain in increasing the bias current and the size of the transistor. The sub–threshold transconductance, g_m and the total gate capacitance, C_{gate}, of the transistor are given as

$$g_m = \frac{\kappa I_d}{U_T}, \;\; C_{gate} \approx C_{ox} W L \tag{11}$$

where I_d is the drain current through the FG transistor; C_{ox} is the oxide capacitance per unit area; W, L are width and length of the transistor. Using the equation for drain current for a transistor operating in saturation , the current at threshold, I_{th}, is given by

$$I_{th} \approx I_{th}^{'} \frac{W}{L}, \;\; I_{th}^{'} \approx \mu_o C_{ox} \frac{U_T^2}{\kappa} \tag{12}$$

where $I_{th}^{'}$ is a constant depending on the process technology. This gives the peak unity-gain frequency,$f_{o,max}$, in sub–threshold using 5 as,

$$f_{o,max} \approx \frac{\kappa I_{th}^{'}}{C_{ox} U_t L^2} \approx \frac{\mu_o U_T}{L^2} \tag{13}$$

As seen, $f_{o,max}$ is independent of the transistor width, W, and threshold current, I_{th}. This value can be obtained when the transistor is operating in sub–threshold region.

This value has striking similarity to it's digital counterpart. The only way to increase this value is by going to smaller technologies i.e. decreasing L, which is similar to improve performance in digital circuits.

2.5.2 Regions of Operation

It is clear from the previous section that the maximum operating frequency is fixed for a particular technology. The region of operating where the devices in the circuit are biased has to be decided based on the design specifications such as area, power and frequency response. In the event that the dominant capacitance is an external capacitance, which is considerably larger than the parasitic capacitance, operating circuits in sub–threshold can be extremely beneficial. Also, operating in sub–threshold works great for low frequency designs, 1kHz to 1MHz. In addition, the transconductance values are higher even in moderate inversion as compared to strong inversion due to the fact that the current-voltage relationship is different than the widely assumed square-law. To summarize, the best way to do power efficient design for particular speeds is:

- Sub–threshold operation for low frequencies (close to peak cutoff for sub–threshold) and cases where the external capacitance is the dominant one.
- Moderate inversion operation for mid-range frequencies, 1MHz - 100MHz.
- Above–threshold operation for high frequencies, close to peak cutoff of the transistor for above-threshold.

To better understand this, consider a ring oscillator. The dominant capacitance in the ring oscillator is the parasitic capacitance as each inverter is driving the next inverter. This implies that the design with minimum W and L will result in the fastest frequency as minimum parasitics. Increasing W of the transistors may give faster driving capacity but it also increases the load proportionately, thus keeping the

operating frequency the same. This shows that increasing the size of the transistor does not always result in a faster design. This becomes really important in array processing where real estate is a big consideration.

Looking at the first option mentioned above, the circuits operating at frequencies between (1KHz - 1MHz) can be easily designed in sub–threshold with close to minimum dimensions, which in turn means minimum parasitics. The reason for this is that the current at threshold is large enough for these devices to encompass the entire frequency range. Thus applications like audio-band processing can be done in the most power-efficient way. This is not usually the practice so far as there hasn't been any need but it becomes extremely important for low–power portable systems. For IF–band applications, designing the transistor to operate in sub–threshold or close to threshold will give the optimum performance in terms of speed for the amount of power consumed.

2.6 Summary

We presented our programmable device element in this chapter. FG device can be used to store charge like an analog memory and can also be used as a signal processing element to design programmable analog systems. As we discussed in this chapter, the performance of FG device when used in signal paths in circuits is comparable to a nominal MOS transistor. We also described our adaptive programming algorithm that can be used to program the FG devices accurately in an array.

We briefly discussed how to design power–efficient systems. As presented, the only real way to increase the maximum operating frequency is by going to smaller technologies. Thus, burning extra power may not always be the best solution to get higher speed performance. This enables the audio- and IF-band systems to be designed in sub–threshold or close to threshold and have the optimal performance. We will now use the concepts presented in this Chapter to design programmable

analog systems to perform signal processing while consuming the optimum power for the desired bandwidth.

CHAPTER 3
VECTOR-MATRIX MULTIPLIER

One of the fundamental operations used in a variety of signal processing applications such as FIR filtering [25], convolution or correlation operations and performing transforms, such as Discrete–Cosine Transform (DCT), is that of vector–matrix multiplication. Current digital realizations of this operation are both area and power intensive for a reasonably sized array, thus making it impractical for large VLSI systems [26]. An analog implementation of such a fundamental operation can help to investigate the feasibility of our hypothesis of power efficient systems for audio, video and IF band signal processing applications. The computation can be done in parallel and faster in analog since the weights stored at each multiplier site saves the fetch time [19, 27]. There have been a number of analog voltage-mode analog implementations for vector–matrix multiplication operation [28, 29]. Previous implementations have used some modification of EEPROM cells [28] or some variation of multiple-input floating-gates for analog storage [29]. The programming schemes used in these implementations were slow and inaccurate. We present a current-mode analog implementation of vector–matrix multiplier using our floating–gate devices and discuss the initial measured results along with the performance. We use our adaptive programming technique that allows for fast and accurate programming [20].

In the next section, we briefly give an overview of previous analog implementations for vector–matrix multipliers. In section 3.2, we present our core multiplier cell. We discuss the basic operation along with the design equations and performance parameters. We present the complete implementation of vector–matrix multiplier (VMM) architecture in section 3.3. Section 3.4 also presents measured results showing the multiplier operation and the measured performance. In section 3.5, we present performance of the designed VMM when used in applications such as Discrete-Cosine

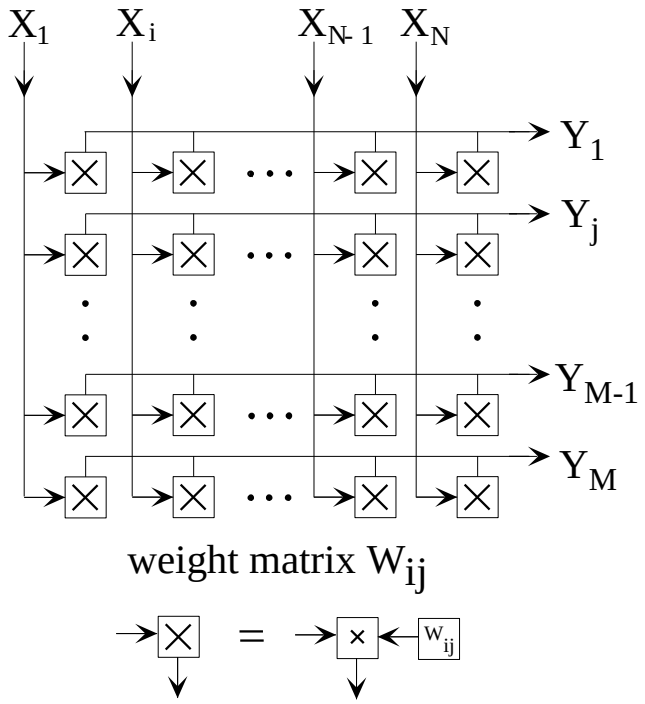


Figure 12. Block diagram of a vector-matrix multiplier.: Schematic of a vector–matrix multiplier that is suitable for an analog implementation, where the input vector, X_i, and output vector, Y_j can be voltages or currents.

Transform (DCT). Section 3.6 summarizes the operation and performance of the designed VMM along with possible applications where this implementation can be used with little modification.

3.1 Overview of Vector-Matrix Multipliers

The basic vector-matrix multiplication operation is defined as sum of products, namely

$$Y_j = \Sigma_i W_{ji} X_i \tag{14}$$

with X_i is a input vector, W_{ji} is a matrix of stored weights and Y_j is the output vector. Figure 12 shows the schematic of a vector–matrix multiplier that is suitable for an analog implementation, where the values X_i, W_{ji} and Y_j can be voltages or currents. The input vector values X_i are multiplied along each column by the stored weight W_{ji} and the results are summed along each horizontal row. The output vector

Y_j are available in parallel in each row. The analog weight matrix values W_{ji} are stored at each multiplier site such that all the multipliers in the array can process in parallel without the necessity to fetch the weight from an external memory.

3.1.0.1 Weight Storage

There have been various proposed implementations for the analog multiplication operation in voltage-mode. In the schematic shown in Figure 12, each multiplier cell require a weight storage mechanism and a multiplication operation. One possible way for having analog storage for the weight values can be achieved using capacitors [30], but these weight values will need to be refreshed after short time. This refreshing operation needed for capacitors use an additional digital memory along with the need to generate analog voltage for the capacitors. For a nonvolatile weight storage, analog electrically erasable and programmable read–only memory (EEPROM) devices can be used that do not need refreshing. Any analog weight value can be stored by programming the threshold voltages of these devices. Previous implementations using some modification of EEPROM cells [28] or some variation of multiple-input floating-gates for analog storage [29] required two gates and two capacitors per cell and in some cases failed to exploit the full benefit of these cells especially accurate programming. In the implementations where these cells were programmed, the schemes used were slow and inaccurate.

3.1.0.2 Multiplier Operation

The multiplier operation for an analog implementation can be obtained using various methods. Size and precision of each of these cells will affect the performance of the vector-matrix multiplier system. A simple analog multiplier circuit that uses MOS transistors in triode [31], is shown in Fig. 13(a) using floating–gate devices. Let us assume that the threshold voltages of $M1$ and $M2$ are $V_{tho} + W_{ji}$ and V_{tho}, respectively. The input vector X_i is applied to the drains and the multiplication is carried out by operating the transistors in triode region. The drain currents for the

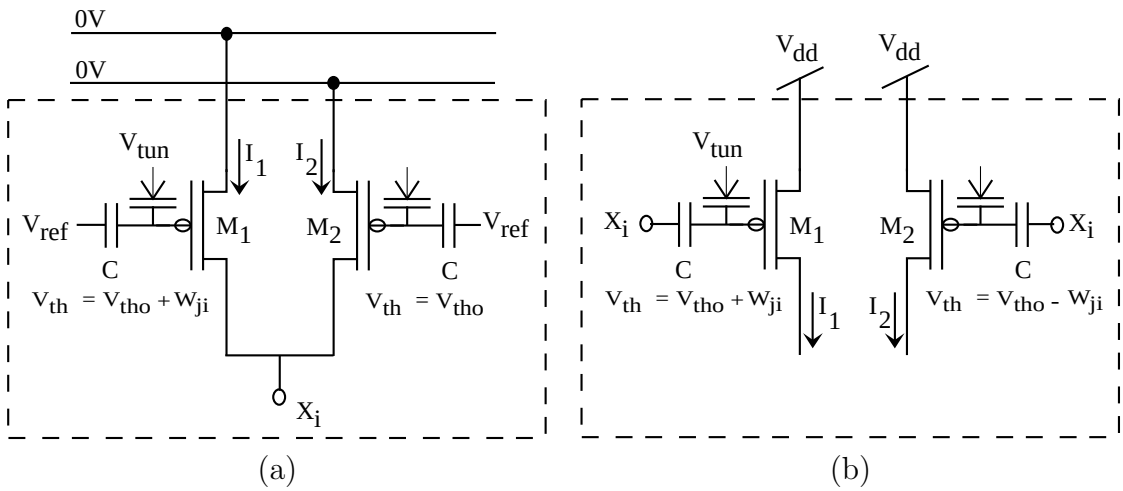


Figure 13. Multiplier cells for voltage–mode VMM implementations.:(a) A MOSFET triode multiplier with floating–gate devices for nonvolatile weight storage. (b) Voltage-mode multiplier cell with MOSFETs operating in saturation. Floating–gate devices are used for nonvolatile weight storage.

two transistors can be approximated as

$$I_1 = \mu C_{ox}\frac{W}{L}[(V_{gate} - V_{tho} - W_{ji})X_i - \frac{X_i^2}{2}] \quad (15)$$

$$I_2 = \mu C_{ox}\frac{W}{L}[(V_{gate} - V_{tho})X_i - \frac{X_i^2}{2}] \quad (16)$$

The product can be obtained by subtracting the two currents as

$$I_2 - I_1 = \mu C_o x\frac{W}{L}(W_{ji}X_i) \quad (17)$$

The disadvantage of the triode multiplier is that any variation of the source of $M1$ and $M2$ would affect the drain-source voltage and thus influence the multiplication result. Another realization of multiplication can be implemented using MOS transistors operating in saturation based on 'quarter-square algebraic identity' that can be written as

$$Y = [(X + W)^2 - (X - W)^2] = 4XW \quad (18)$$

This implementation needs to first add and subtract input signal X and W. The realization of such a multiplier is presented in [32]. Although transistors are operating in saturation, it requires too many transistors (at least 12 transistors) and is

thus not suitable for array implementation. [32]. Similar to Fig. 13(a), we can use MOS transistors in saturation by applying the input signal X_i to the gate of the two transistors as shown in Fig. 13(b). The drain currents are given by

$$I_1 = \mu C_{ox} \frac{W}{L} (X_i - (V_{tho} + W_{ji})^2) \quad (19)$$

$$I_2 = \mu C_{ox} \frac{W}{L} (X_i - (V_{tho} - W_{ji})^2) \quad (20)$$

The product can be obtained by subtracting the two currents as

$$I_2 - I_1 = 4\mu C_o x \frac{W}{L} (X_i - V_{tho}) W_{ji} \quad (21)$$

Implementation using this multiplier cell and dual-input floating-gate MOS that requires two capacitors per cell was shown in [19]. This implementation, although compact, has offsets in the final results that have to be corrected off-chip and does not provide a fully–differential operation. One of the major limitation in all the previous voltage-mode implementations is that the maximum linearity available is limited up to power supply rails. A voltage–mode implementation of analog multiplier using floating–gate devices was developed in our group [6]. All of these implementations operated at slow speeds and had high power consumption, which can be a limiting factor in some of the portable high-speed applications like video processing.

3.2 Core Programmable Multiplier Cell

We designed a current–mode implementation of the vector matrix multiplication (VMM) operating in sub–threshold regime thereby achieving low-power operation and high linearity. The addition operation is done using KCL and hence, does not dissipate any additional power when compared to the digital approaches. We will now discuss the basic operation of our core multiplier cell along with design equations that govern the performance of our multiplier cell.

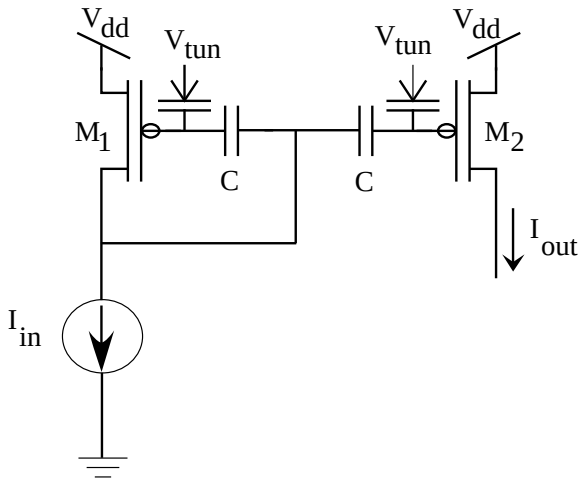


Figure 14. Core multiplier cell: Circuit schematic showing the core current–mode multiplier cell. This core cell can easily be made fully–differential, as will be shown, and is used to implement the final VMM system.

3.2.1 Weight Storage

Figure 14 shows the circuit schematic of our core current-mode multiplier cell that is used to implement the VMM system. The multiplier cell makes use of a floating–gate current mirror with the two floating-gates programmed to different charges. Both transistor M_1 and M_2 are floating–gate nonvolatile devices that are used to store the weight. Our adaptive programming technique allows for fast and accurate programming of these floating–gate devices [20]. In our implementation, single floating–gate device is used as a signal–processing element for the multiplication along with nonvolatile weight storage.

3.2.2 Basic Multiplier Operation

We now consider the operation of this multiplier in both sub-threshold and above-threshold regimes, and compare their performance.

3.2.2.1 Sub-threshold Region Operation

Consider the floating-gate transistors M1 and M2 (refer Fig. 14) that are programmed to different floating-gate charges $V_{charge,1}$ and $V_{charge,2}$. The drain current of M1 and

M2 using 1, neglecting early effect are given by,

$$I_{in} = I_o exp(\frac{-\kappa(V_{fg} + V_{charge,1}) + V_s}{U_T}) \tag{22}$$

$$I_{out} = I_o exp(\frac{-\kappa(V_{fg} + V_{charge,2}) + V_s}{U_T}) \tag{23}$$

Therefore, the multiplier weight, W_{21}, is given by,

$$W_{21} = \frac{I_{out}}{I_{in}} = exp(\frac{-\kappa(V_{charge,2} - V_{charge,1})}{U_T}) \tag{24}$$

Different multiplication weights can be implemented by programming the difference in the floating-gate charges of transistors M1 and M2. Theoretically, the above weight equation translates to decades of linearity as long as the two transistors remain in the sub-threshold region of operation. However, 24 is derived under the assumption that κ does not vary with surface potential and hence the programmed floating-gate charge. Fig. 15 shows the measured plot for κ for different programmed currents and, hence, different V_{charge} and clearly demonstrates the change of κ with floating-gate charge. Incorporating this second order effect, the weight is now given by,

$$W_{21,act} = exp(\frac{-(\kappa_2 V_{charge,2} - \kappa_1 V_{charge,1})}{U_T}) exp((\kappa_2 - \kappa_1)\frac{\Delta V_{fg}}{U_T}) \tag{25}$$

The dependence of the weight on the change in the floating-gate voltage limits the linearity of the multiplier structure. A possible solution to increase the linearity would be to program the two floating-gate transistors relatively close to each other such that their κ's are almost equal. This approach will yield fractional weights that can easily be amplified in later stages, if needed.

3.2.2.2 Above-threshold Region Operation

The drain currents of M1 and M2 (as shown in Fig. 14) in saturation, are given by,

$$I_{in} = \frac{\kappa K}{2}(V_{sfg} - V_{th,1})^2 \tag{26}$$

$$I_{out} = \frac{\kappa K}{2}(V_{sfg} - V_{th,2})^2 \tag{27}$$

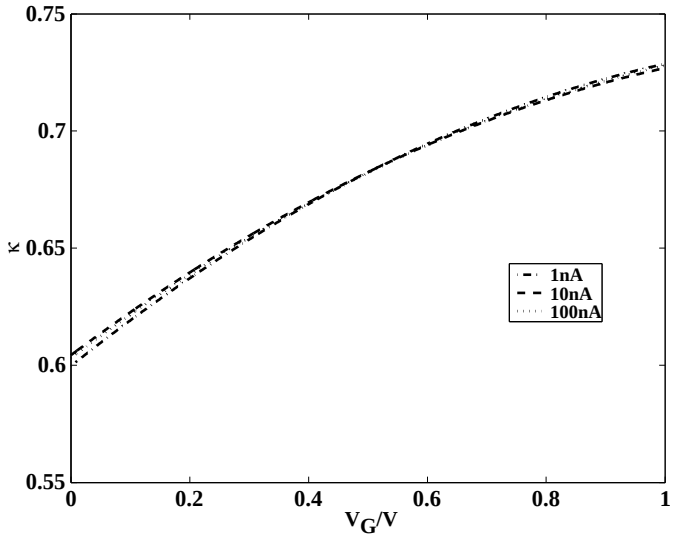


Figure 15. Variation of κ when same current is programmed for different gate voltages: Plot showing variation of κ when the same current is programmed at different gate voltages. As can be seen, the value of κ changes with different surface charge and is not constant for all values of gate voltage.

Now, the multiplication weight, W_{21}, is given by,

$$W_{21} = \frac{I_{out}}{I_{in}} = \frac{(V_{sfg} - V_{th,2})^2}{(V_{sfg} - V_{th,1})^2} \tag{28}$$

A change in the input current (ΔI) creates a change in the gate voltage (ΔV_g) that in turns creates a change in the floating gate voltage (ΔV_{fg}). Now, the multiplication weight becomes,

$$W_{21,act} = \frac{(1 + \frac{\Delta V_{fg}}{V_{sfg} - V_{th,2}})^2}{(1 + \frac{\Delta V_{fg}}{V_{sfg} - V_{th,1}})^2} \times W \tag{29}$$

This can also be expressed as,

$$W_{21,act} = \frac{(1 + x\sqrt{W_{21}})^2}{(1 + x)^2} \times W_{21} \tag{30}$$

where, x is given by, ΔV_{fg} / $(V_{sfg} - V_{th,1})$. For a given signal swing and a multiplication weight W, the smaller the value of x the closer the weight, $W_{21,act}$, is to W. This translates to a limited linearity coupled with a higher power dissipation. Hence, for the proposed multiplier, operation in sub-threshold proves to be beneficial both from a linearity and power consumption point of view.

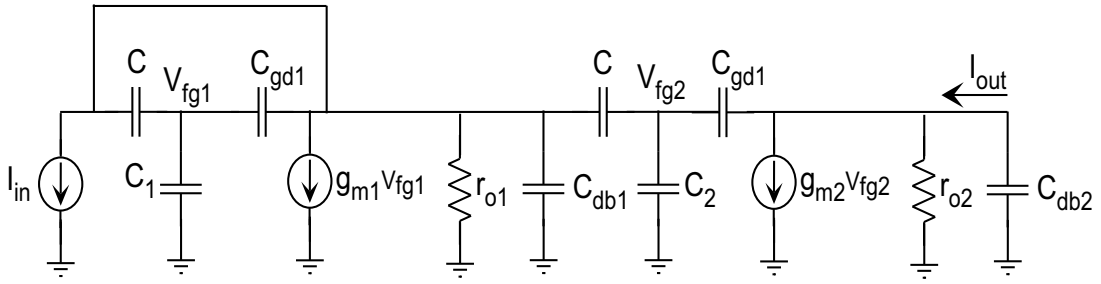


Figure 16. Small-signal model for the current–mode multiplier cell: This simplified model can be used to develop an understanding of the performance such as speed, SNR of the multiplier cell.

3.2.3 Frequency Performance

Figure 16(b) shows the small-signal equivalent model of the core multiplier element. The capacitor C_1 shown in the figure is a combination of a number of parasitics and is given by,

$$C_1 = C_{gs1} + C_{gb1} + C_{tun} \tag{31}$$

where, C_{gs1} represents the floating gate-source capacitance of transistor M1, C_{gb1} represents the floating gate-bulk capacitance of M1 and C_{tun} represents the tunneling capacitance. It should be noted that to a first approximation, C_1 is dominated by the floating gate-source capacitance C_{gs1}. Also, the capacitance C_2 is the analogous lumped capacitance at the floating–gate of M2. Assuming that gate-drain overlap capacitance, C_{gd}, is small when compared to C, floating-gate voltage V_{fg1} and V_{fg2} can be approximately written as,

$$V_{fg1} = \frac{(C_{gd1} + C)}{(C_{gd1} + C + C_{gs1})} V_x \approx \frac{C}{C_{T1}} V_x \tag{32}$$

$$V_{fg2} \approx \frac{C}{C_{T2}} V_x \tag{33}$$

where

$$C_{T1} = (C_{gd1} + C + C_{gs1}), \;\; C_{T2} = (C_{gd2} + C + C_{gs2})$$

Applying KCL at node V_x,

$$sC(V_x - V_{fg1}) + sC(V_x - V_{fg2}) + I_{in} + g_{m1}V_{fg1} + sC\frac{C_{gs1}}{(C + C_{gd1})} = 0$$

Using these equations and neglecting the effect of C_{gd}, output current, I_{out}, is given by

$$I_{out} \approx g_{m2}V_{fg2} = g_{m2}\frac{C}{C_{T2}}V_x \tag{34}$$

When using floating–gate devices, the (W/L)'s of the two devices are identical. Thus, the parasitic capacitances for the two devices are equal. This gives the current gain as

$$\frac{I_{out}}{I_{in}} \approx \frac{g_{m2}}{g_{m1}}\frac{1}{s(2C_{gs1}) + g_{m1}} \tag{35}$$

The above expression is a first-order response of the circuit. Including the effect of the C_{db2} and the output transistor into which I_{out} is flowing will give the secondary non-dominant high-frequency poles. Along with that, there is a zero at the output due to the C_{gd2} that can be eliminated by the use of a cascode transistor, as used in our implementation.

3.2.4 Signal-to-Noise of Multiplier Cell

In this sub-section, the signal-to-noise ratio (SNR) of a simple floating-gate current mirror is derived and the implications of the result is analyzed. Figure 16(a) shows the core programmable floating–gate multiplier cell along with the small-signal representation as shown in Fig. 16(b). The f_{-3dB} frequency and the noise bandwidth of the circuit can be derived from (35) and are given by,

$$f_{-3dB} = \frac{1}{2\pi} \cdot \frac{g_{m1}}{2C_{gs1}} \tag{36}$$

$$NoiseBandwidth = \frac{\pi}{2} \cdot f_{-3dB} = \frac{1}{4} \cdot \frac{g_{m1}}{2C_{gs1}} \tag{37}$$

The total noise spectral density at the output is equal to the sum of the noise contributions of each of the two transistors.

$$\frac{i_o^2}{\triangle f} = 4KT\frac{2}{3}(g_{m1} + g_{m2}) \tag{38}$$

Referring the noise back to the input we get,

$$\frac{i_{in}^2}{\triangle f} = \frac{8}{3}KT(g_{m1} + g_{m2})\frac{g_{m1}^2}{g_{m2}^2} \tag{39}$$

Using the above expression and the expression for the noise bandwidth, we find the total input referred rms noise to be,

$$i_{in,rms} = \sqrt{\frac{KT(g_{m2} + g_{m1})}{3C_{gs1}}} \cdot \frac{{g_{m1}}^{3/2}}{g_{m2}} \tag{40}$$

At this point, the SNR of the floating-gate current mirror can be calculated by assuming that the given current mirror has a bias current of I_{bias} flowing through it. The rms value of the full-scale input signal then becomes,

$$i_{sig,rms} = \frac{I_{bias}}{2\sqrt{2}} \tag{41}$$

The SNR for the core cell in sub-threshold turns out to be,

$$SNR = \frac{g_{m2}I_{bias}}{{g_{m1}}^{3/2}}\sqrt{\frac{3C_{gs1}}{8kT(g_{m1} + g_{m2})}} \tag{42}$$

The implications of 42 can be analyzed from two regimes of operation: (a) Sub-threshold and (b) Above threshold. In sub-threshold operation, the transconductance is directly proportional to the current and so (42) simplifies to,

$$SNR = nU_T\sqrt{\frac{3C_{gs1}}{8KT[1+n]}} \tag{43}$$

where, n denotes the ratio of the drain currents of $M2$ and $M1$. For above-threshold operation, the transconductance is proportional to the square-root of drain current and so, (42) simplifies to,

$$SNR = \frac{\sqrt{nI_{bias}}}{\sqrt{2\mu_p C_{ox}(\frac{W}{L})_1}}\sqrt{\frac{3C_{gs1}}{8KT[1+\sqrt{n}]}} \tag{44}$$

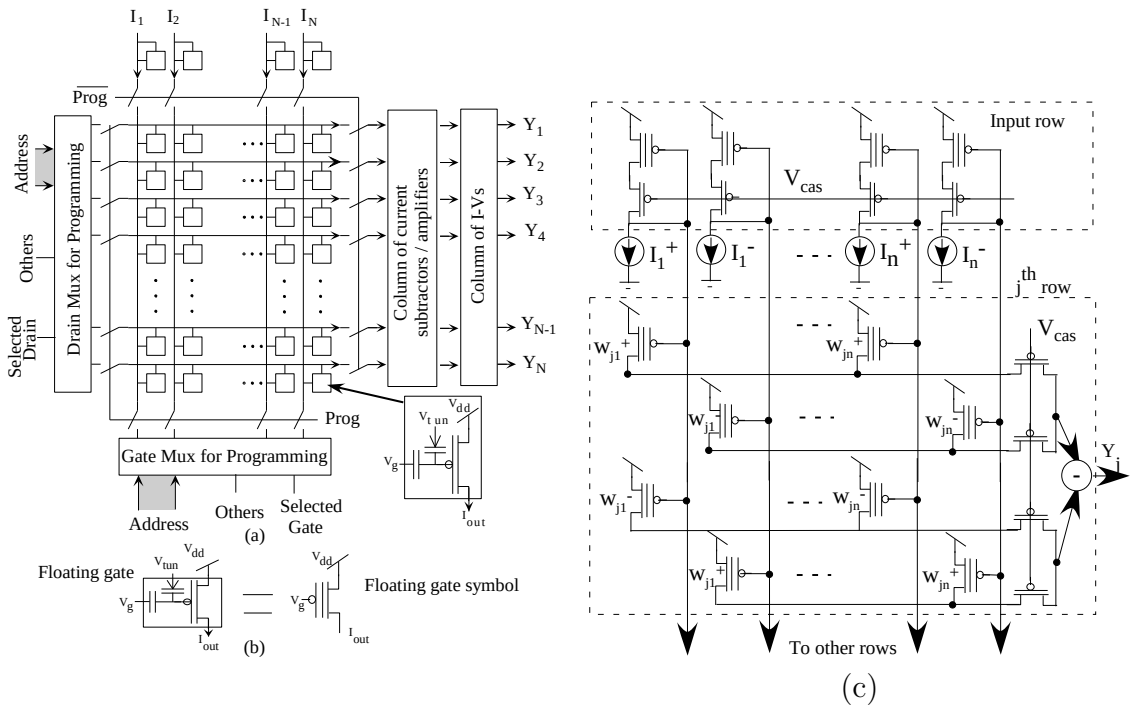


Figure 17. Block diagram and circuit schematic of vector-matrix multiplier (VMM): (a) The chip consists of a 128x32 array of floating–gate vector matrix multiplier elements, peripheral digital control for isolation of floating–gate elements during programming, and current amplifiers. (b) Symbol used for a floating–gate (FG) device. (c) Circuit schematic showing the j^{th} row for a fully–differential current–mode vector–matrix multiplier;

Thus, for a sub-threshold operation, the SNR is independent of the current level while for above-threshold operation, the SNR is directly proportional to the square-root of bias current. In both cases, the SNR improves with a larger C_{gs1} or larger transistors. This is analogous to the KT/C noise of a simple RC network and presents a direct tradeoff between the SNR of the multiplier and the cell area of the multiplier.

3.3 Multiplier Implementation

Figure 17(a) shows the block diagram of our programmable current-mode VMM architecture using floating-gate (FG) elements. Also, the addition operation is done using KCL and hence, does not dissipate any additional power when compared to the digital approaches. The exponential I-V relationship of transistors operating in sub-threshold provides a logarithmic compaction that increases the linearity of our

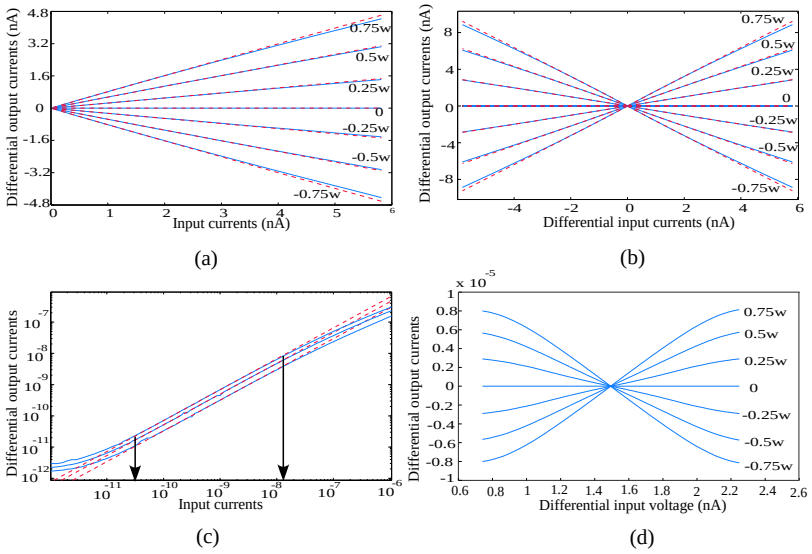


Figure 18. Experimental results for current–mode multiplier: (a) Plot of measured differential output current vs. input current on a linear scale, for two-quadrant configuration; (b) Measured differential output current vs. differential input current for four-quadrant configuration; (c) Measured differential current output vs. differential input voltage for a voltage–mode configuration; (d)Plot showing the limits of linearity for the current–mode configuration for the two-quadrant configuration. ***In these plots the solid lines are measured data while the dashed lines are ideal fits***

multiplier architecture as compared to a voltage-mode technique. Using this current–mode implementation in sub-threshold gives the most bandwidth for a given power dissipation provided the dominant capacitances are intrinsic to a transistor. The proposed architecture provides for programmable, non-volatile weight storage through the use of floating–gate MOSFETs operating in the signal path. Our adaptive programming technique allows for fast and accurate programming of these floating–gate MOSFETs using standard CMOS devices [17]. Floating–gate MOSFETs are programmed by isolating each individual transistor by means of digital logic consisting of switches, decoders and multiplexors. It should be noted that the entire digital logic required for programming occupies only 3 % of the total chip area. Also, to aid in measurement, the output currents are amplified and then converted into a voltage using linear I-V Converters.

Figure 17(b) shows a detailed circuit schematic of the VMM system. Our VMM

chip affords the flexibility of configuring the system as either a two-quadrant or a four-quadrant multiplier for both positive and negative weights. This can be achieved by using the inputs differentially or in a single-ended fashion. For the two-quadrant configuration, the common–mode cannot be intrinsically rejected on–chip. Different rows were programmed to different weights and all the weights in one particular row were programmed identical. Fig. 18(b) and 18(c) demonstrate the functionality as a two-quadrant and four-quadrant multiplier respectively. Four-quadrant operation eliminates output DC offsets on-chip and even-order harmonics, and helps improve linearity. This is evident from Fig. 18(c) and (3).

$$\begin{aligned} Y_j &= \Sigma[(w_{ji}^{+} - w_{ji}^{-})(\Delta I_i^{+} - \Delta I_i^{-}) \\ &\quad + \frac{(w_{ji}^{+} - w_{ji}^{-})^3((\Delta I_i^{+})^3 - (\Delta I_i^{-})^3)}{3}] \end{aligned} \tag{45}$$

The linear range of the multiplier can be estimated from Fig. 18(e) that shows the differential output current vs. the input current for various positive weights. The linearity is measured to be greater than two decades, beyond which the multiplier deviates from the ideal linear curve with an error that is higher than 2.5%. As explained earlier, this linearity limitation is partly due to the difference in κ between identical transistors programmed to different currents and the variation of κ with the gate voltage. This effect can be alleviated by programming the elements relatively close to each other. Fig. 18(e) also emphasizes the point that a current-mode implementation gives decades of linearity in signal swing that is especially hard to obtain in voltage-mode circuits without consuming more power. For instance, in [19], a linear range of 1V - 4V is obtained at the expense of 0.39mW of power dissipation. Figure 18(d) shows the linearity plot of a voltage-mode multiplier that we fabricated in 0.5μm CMOS process. As can be seen, the circuit operated in above-threshold and absorbed mW of power to give barely 1V linearity. This linearity can be improved using some techniques but it comes at an expense of reduced speed or more power at

same speed. In our implementation, the DC level of input current determines both the speed and the power dissipation and can be programmed to any desired value.

In a floating-gate device, the output impedance is degraded primarily due to the drain voltage (V_d) variation coupling onto the floating-gate node through C_{gd} rather than Channel Length Modulation; cascoding helps in reducing the C_{gd}-coupling effect by making the drain of the floating-gate a low impedance node while maintaining a high impedance at the output. This also helps improve the distortion due to the isolation from the output signal variations. A cascode transistor was added for each row of n floating–gate devices with a size n times that of a single floating–gate device. The cascode transistor also helps to reduce the effect of the line capacitance on the frequency response. The pole frequency at the source of the cascode for a n–element row is given as

$$P_{cas} \approx \frac{g_{m,cas}}{C_{tot}} \tag{46}$$

where

$$C_{tot} = (C_{gs,cas} + C_{sb,cas} + nC_{db} + C_{line})$$

The magnitude of this pole is relatively close to the input pole and thus, affects the frequency performance directly. A possible way to increase the magnitude is to add an auxiliary DC current to the cascode to increase the transconductance. This may not be necessary as long as the Q of the system is less than 0.5 because the phase at unity gain frequency for this open loop system will not affect the performance. Also, the cascode transistors can be used as switches in the program mode to better isolate the elements and thus, serve a dual purpose. For the reasons discussed, we will use a cascode device for each row or column of floating–gate devices in most of our system implementations.

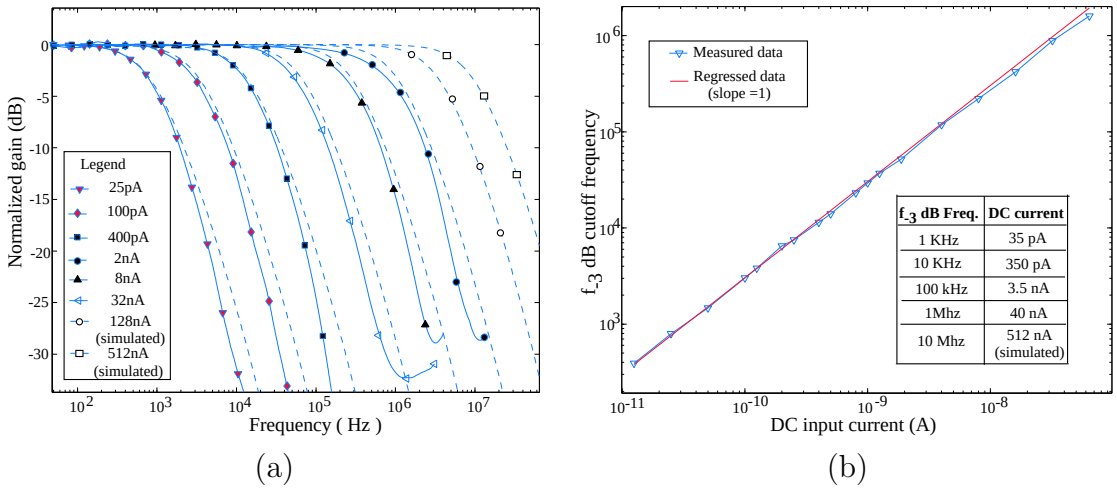


Figure 19. Frequency response of vector–matrix multiplier (VMM): (a) Plot of frequency response of current mode multipliers. The solid lines represent measured data while dashed lines represent simulation results. (b)Variation of f_{-3dB} cut–off frequency vs. DC input current (per FG device) is plotted. For subthreshold currents a linear relationship is observed, as expected. The table shows the measured DC input current (per FG device) required for various f_{-3dB} cut–off frequency.

3.4 Experimental Results and Discussions

A custom PCB was fabricated to perform speed measurements for low input currents. Fig. 19(a) shows the measured and simulated frequency response for different DC input currents. The measured corner frequencies (f_{-3dB}) match closely to the simulated results. The plot shows that the VMM would easily operate up to 10MHz if it was not limited by the frequency response of the I-to-V converter (Bandwidth =

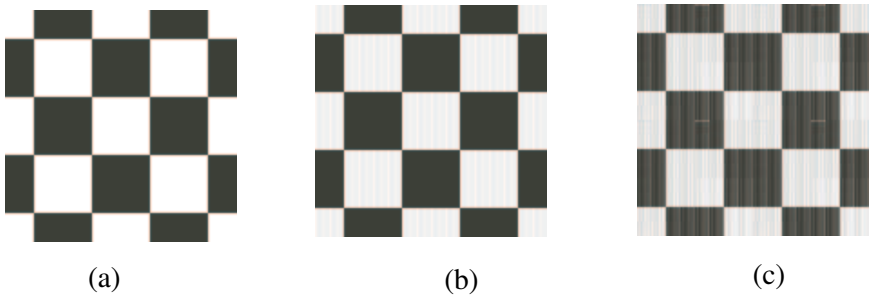


Figure 20. 8x8 block DCT of a 128x128 image: (a) Original input image; (b) Image after inverse DCT, when block matrix transformation was performed off–chip, using the measured weight matrix from the VMM chip. (c) Output of the VMM chip (after inverse DCT) for 8x8 block transform that was performed on–chip.

Figure 21. Die Photograph of the chip: The VMM chip consists of a 128x32 array of floating–gate elements, current amplifiers, and peripheral digital control for isolation of floating–gate elements during programming.

5MHz) at the output. Fig. 19(b) shows a plot of measured corner frequencies with the input DC bias current on a log-log scale. The data points follow a straight line with a slope of 1 as expected in sub-threshold. The deviation for higher current levels is due to the transistor moving from sub-threshold regime to the above threshold region. The bias currents required for a bandwidth of 1MHz and 10MHz are 40nA (measured) and 512nA (simulated), respectively for each FG device.The VMM chip required 531nW/MHz (from Fig. 19(b)) for each differential cell clearly demonstrating the speed vs. power tradeoff. The DC bias current however can be set solely on the basis of speed requirements as the Signal-to-Noise Ratio (SNR) is independent of the input DC bias level. The SNR however is directly proportional to the Gate-Source Capacitance (C_{gs}) and can be increased at the expense of chip area.

Table 1 summarizes the performance of our VMM along with that of [19]. As can be observed, the proposed architecture is both power and area efficient. Fig. 21 shows the micrograph of the VMM chip that was fabricated in a 0.5μm N-well CMOS process.

3.5 Application: Block transform of images and FIR filtering

The VMM chip can be used for applications like audio and video processing. The VMM architecture was configured to perform real–time block matrix transforms of

Table 1. Performance Summary for Vector-Matrix Multiplier

Parameter	**Proposed VMM**	**VMM in [19]**
Technology	0.5μm N-Well CMOS	1.5μm single poly CMOS/EEPROM
Power Supply	3.3V	5V
FG Dim.(W/L)	18λ / 4λ	N/A
Array size	128×32	16×16
Chip area	0.83mm^2	1mm^2
Programming % error	$< \pm 0.2\%$	<10mV
BW/power per cell	531 nW/MHz	N/A
Linearity	$>$ 2 decades	3V
Power per cell	7.2 μW @10MHz	0.39mW @60KHz
Programming scheme	Hot–electron injection and Tunneling	Electron Tunneling
Programming Time per W_{ji}	1mS	100mS

input images in a row–parallel manner as proposed in [33]. The weights were programmed to be the DCT kernel. Fig. 20(a) shows the 128x128 image that was placed as an input to the chip. To estimate the performance of the VMM, the programmed weights were first measured and the block DCT (8x8) was performed off-chip. Fig. 20(b) shows the image obtained after inverse transformation. Next, the block transform was performed on-chip and the result is shown in 20(c). It can be observed that the results for part (b) and (c) are similar thereby demonstrating the usefulness of our VMM architecture. The distortion observed in both the images are due to the programming accuracy limitations (0.2% error).

3.6 Summary

In this chapter, we presented a programmable fully–differential current–mode VMM architecture. The architecture is suitable for low voltage, low power applications and has a bandwidth-to-frequency ratio of 531nW/MHz per differential multiplier cell. A linearity of over two decades has been reported for the multiplier. As an application of the VMM, a block matrix transform (DCT) operation has been demonstrated with

good results. For a bandwidth of less than 10MHz, this architecture is capable of performing 1 million Multiply-Accumulate (MAC) operations/0.27μW as compared to a commercially available DSP (TMS320C55105x series), which gives 1 million MAC/0.25mW. This fundamental signal–processing system is one example showing the huge power advantage when using analog techniques as oppose to the popular digital approaches. The sub–threshold operation of the system further enhances the performance in terms of power due to maximum g_m/I ratio. These techniques can be extended to variety of other signal–processing systems, where power consumption is an important specification.

In the next chapter, we extend our programmable analog signal–processing techniques to one of the building blocks in any transceiver chain: modulator/demodulator block. We present a programmable analog modulator/demodulator that can be used for a variety of communication schemes along with power–efficient operation.

CHAPTER 4
PROGRAMMABLE ANALOG MODULATOR

As is evident from the previous chapter, floating–gate circuit techniques promise to give power efficient and area effective analog solutions for systems requiring portability and systems performing array signal processing. In this chapter, we present design of a generalized programmable analog modulator/demodulator for any arbitrary communication scheme to demonstrate how floating–gate elements can be used to implement array signal processing systems such as radar and digital signal processing.

The explosive growth of wireless and signal processing applications has resulted in an increasing demand for such systems with low cost, low power consumption, and small die area. To meet this demand, much work is focused on, and has recently demonstrated, fully integrated single-chip systems in low–cost CMOS processes. With the integration of these high–performance systems, power consumption becomes a critical design specification. An IF band signal processing system typically requires the use of an array of DSPs operating in parallel to meet the speed requirements [1, 34]. Figure 22 shows the block diagram of a generalized transceiver from the antenna to all the way down to baseband. This is a power intensive approach and makes use of certain communication schemes impractical in portable systems especially when these systems have to support more applications with same limited power budget. The front-end ADC and back-end DAC converters required in these systems become expensive and power hungry when the signal is of wideband nature and greater resolution is required [35, 36].

Recent focus has been in processing signals as much as possible in the analog domain before converting them into digital. Figure 23 shows the transceiver system block diagram based on the Cooperative Analog/Digital Signal–Processing (CADSP)

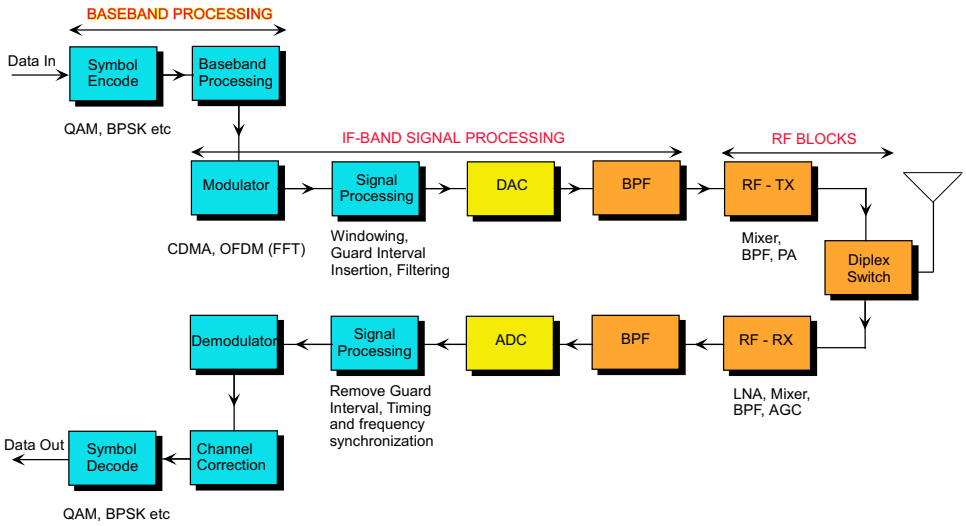


Figure 22. Block diagram for a generalized transceiver system.: Transceiver block diagram showing the RF front–end to all the way down to baseband DSP processor. Purely analog blocks are color coded as orange with digital blocks as blue and the mixed–signal blocks such as ADCs and DACs as yellow.

approach. The motivation is to use both analog and digital approaches together to get maximum power–efficiency while getting comparable performance. One of the major building blocks, as can be seen from Fig. 23, to make this approach practical for real systems is an analog modulator/demodulator system.

We propose a programmable analog arbitrary waveform generator that can be used for a variety of signal processing applications [37]. The waveform generator is fully programmable through use of floating-gate MOS transistors. The programmable arbitrary waveform generator can be used as a building block for the Programmable Analog Modulator/Demodulator (PAMD). PAMD can be one of the fundamental blocks in the transceiver, as shown in Fig. 23, enabling a lot of other signal processing functions in the analog domain [4, 6, 7, 8, 9, 10, 11]. This approach is power and area efficient as compared to complex DSPs and relaxes the requirement on the design of data converter specifications.

The proposed PAMD implementation can be used in various communication

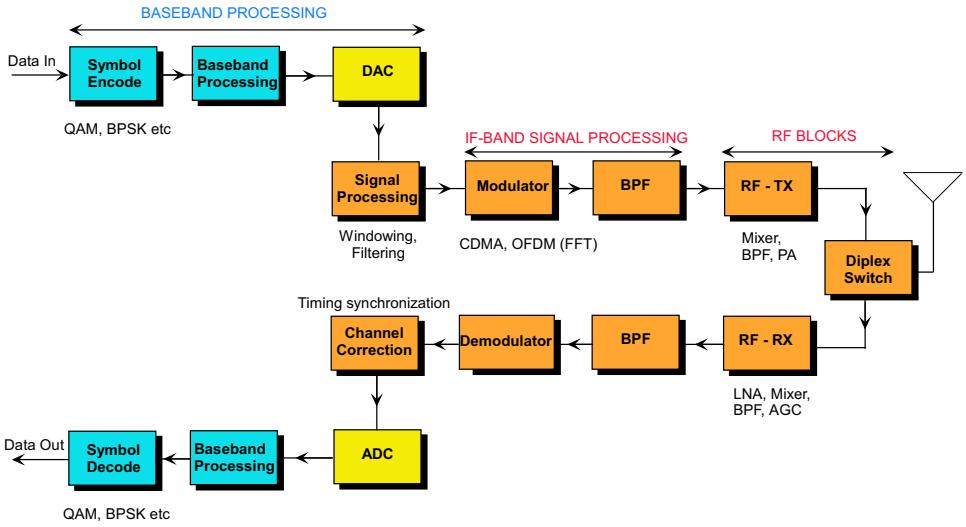


Figure 23. Block diagram for a CADSP transceiver system.: Transceiver block diagram showing the RF front–end to all the way down to baseband DSP processor. The main idea of CADSP is to perform as much signal–processing in analog as possible before converting the signal to digital. This approach promises a huge advantage in terms of power that is becoming a critical specification for portable applications. Purely analog blocks are color coded as orange with digital blocks as blue and the mixed–signal blocks such as ADCs and DACs as yellow.

schemes such as Orthogonal Frequency Division Multiplexing (OFDM) and radar signal processing [35]. The biggest advantage comes from the fact that the waveforms generated can be arbitrary and are programmable. In section 4.2, we discuss the programmable analog waveform generator. Section 4.3 presents the PAMD implementation using the programmable waveform generator along with measured experimental results as modulator and demodulator. We conclude in section 4.4 with possible applications of the presented architecture.

4.1 Overview of Modulator and Demodulator Systems

In this section, we will briefly elaborate on the existing modulator and demodulator architectures for certain communication schemes. Most of these systems are limited by the signals that can be generated with ease for modulation and demodulation. Currently, most of these systems are implemented using digital signal processors

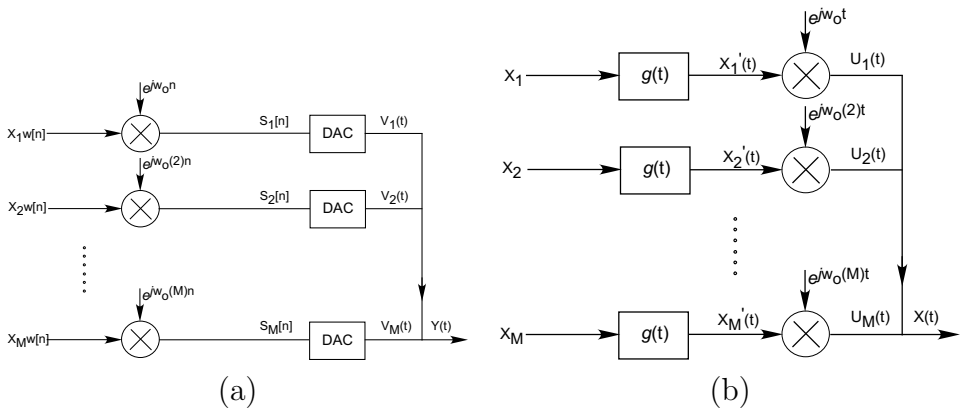


Figure 24. Analog representation and digital implementation of a OFDM transmitter: (a) Digital implementation of OFDM transmitter requiring DACs for each channel and an FFT computation block. (b) Analog representation of OFDM transmitter. Any pulse shaping function, $g(t)$, can be used in this representation.

(DSPs) [38] and can be power hungry. To further illustrate, let us consider an Orthogonal Frequency Division Multiplexing (OFDM) modulator. OFDM is a modulation technique suggested for use in cellular radio, digital audio and video broadcasting. OFDM uses a number of orthogonal sub-carriers for modulation to transmit data in parallel. The main advantage of using OFDM is that modulation and demodulation can be achieved using Fast Fourier transform (FFT). Figure 24(a) shows one such digital implementation of an OFDM transmitter [35]. This implementation is based on the analog representation, shown in Fig. 24(b), and can not be implemented for any pulse shaping filter in analog domain as explained in [35]. In terms of circuit implementation, one DAC is required for each channel along with the FFT computation that is to be performed with the sub-carriers. The DACs have to operate on the carrier modulated signal and has to be at least twice as fast as the highest carrier (around 30 MHz). This can be hard to design in a power and area efficient way if the DAC has to have a reasonable bit-resolution as well. The same problem occurs on the receiver-end where wideband ADCs are required for the received signal.

In digital implementations, the power consumption of these systems is often lowered at the expense of lower quantization or by limiting the transmitted symbols to

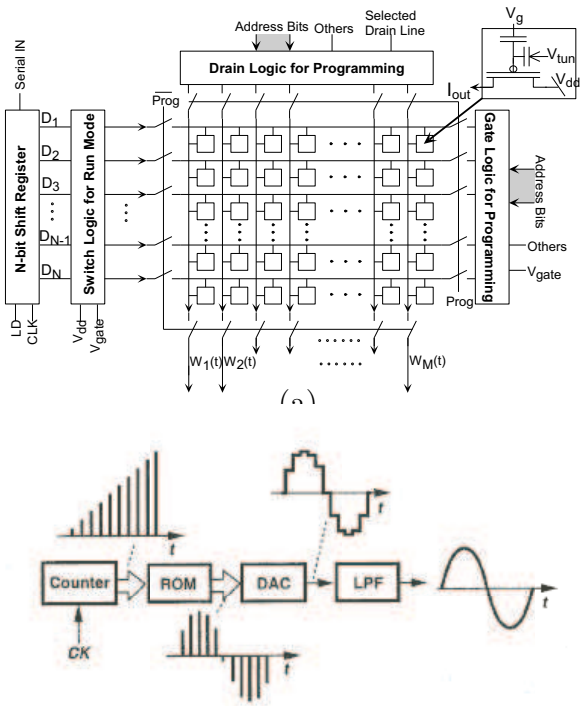


Figure 25. Arbitrary waveform generator (AWG) using floating-gate transistors.: (a) Analog implementation of a waveform generator using floating-gate devices. This architecture is similar to a direct digital frequency synthesizer implementation with floating-gates acting as analog memory cells. (b) Simple block diagram of a DDS system that is used to digitally synthesize a sine wave. [40]

a QPSK constellation. This results in reducing the DAC/ADC resolution along with the lower precision of FFT/IFFT computation. A multi-band OFDM system designed for a single analog receiver chain to simplify the design a lot consumes anywhere from $155mW$ to $170mW$ depending on the data rate [39]. Along with lower resolution, the power budget and difficulty in implementation also limits the variety of communication schemes and coding that can be used for portable applications. One such popular scheme that is often used in optical communication is chirped return–to–zero pulse modulation. This scheme has analog phase modulation across the pulse and improves robustness to non–linear distortions from long transmissions. The biggest issue in using this scheme is to generate these chirped return–to–zero signal for modulation with ease and low–power consumption in standard CMOS technology.

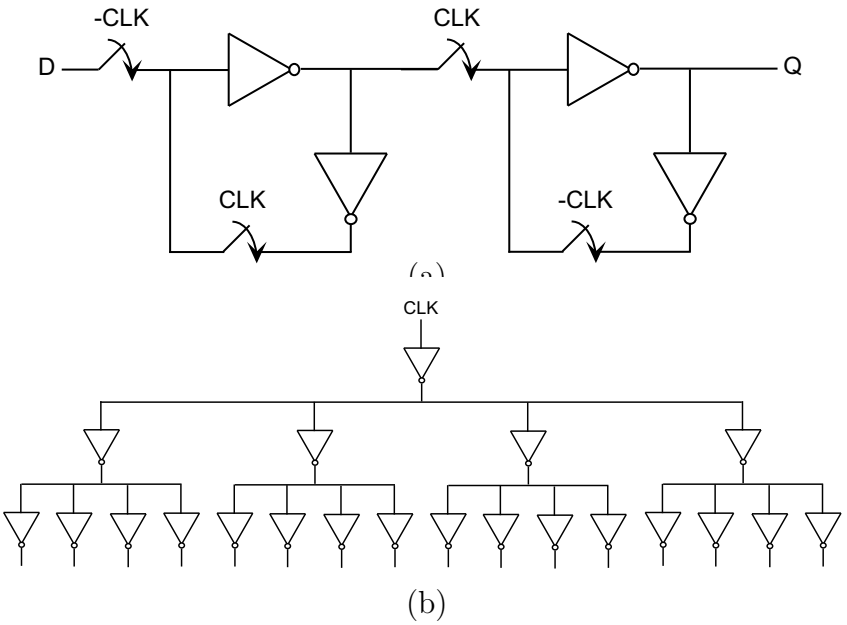


Figure 26. Implementation of AWG: (a) Circuit schematic for D-flip flop (DFF) that was used to implement the shift register. The CLK and -CLK should arrive at all DFF blocks simultaneously. (b) Tree structure buffering and routing of CLK and -CLK to minimize an time and phase difference between different blocks for CLK and -CLK.

We will now present the details of our proposed analog modulator/demodulator systems to address some of the issues mentioned above. As will be seen, the proposed system is fully programmable and thus, eliminating one of the biggest limitations of analog signal processing systems. In the next section, we will discuss our arbitrary waveform generator before discussing the complete PAMD system.

4.2 Arbitrary Waveform Generator

Figure 25(a) shows the block diagram of the waveform generator that is used in the proposed PAMD implementation. This architecture is similar to that of a direct digital frequency synthesizer (DDS) implementation as shown in Fig. 25(b) [40]. The basic idea in DDS is to generate the signal in the digital domain and then utilize D/A conversion and filtering to reconstruct the waveform in analog domain. In the proposed waveform generator, all rows in the waveform generator consist of floating-gate MOS transistors that can be programmed to any analog value. Each floating-gate

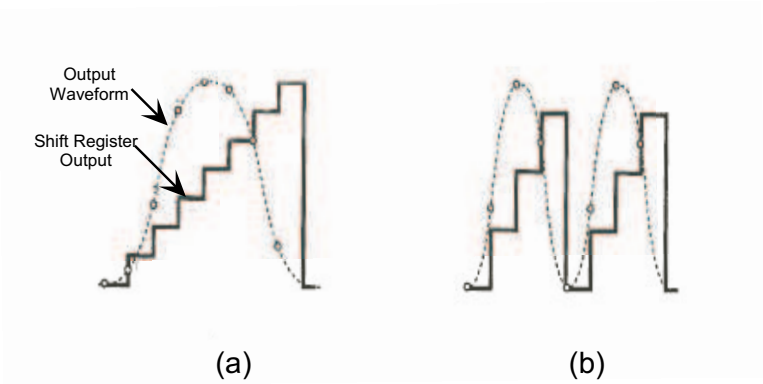


Figure 27. Output waveform generation for a 8-element column [40]: (a) Generation of a sine–wave with eight elements in the floating–gate column. The output of the shift register is shown with the waveform output. (b) The same number of elements can be used to generate sine–wave with twice the frequency as in (a). This can be achieved by programming two cycles instead of one complete cycle.

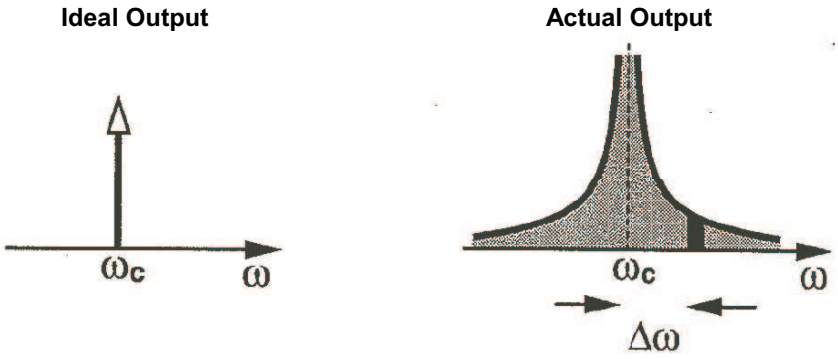


Figure 28. Output spectrum of a PAMD system: Output spectrum of a ideal and actual generated waveform for a PAMD system.

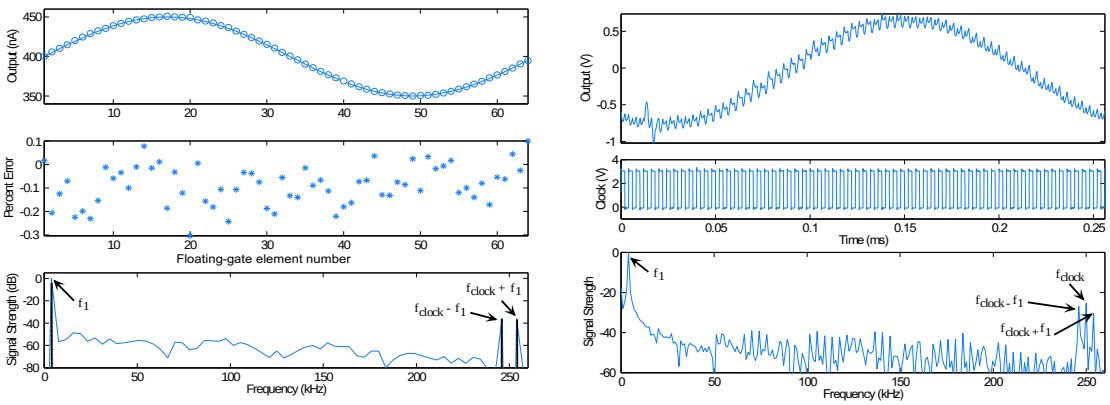


Figure 29. Waveform generator measurements: (a) Measurement showing the output waveform when a $100nA_{pp}$ sine wave is programmed riding on a $300nA$ DC current. Each row has 64 floating-gate elements.. (b) Measurement showing the output waveform when a clock of $250kHz$ is applied to the waveform generator programmed with the sine-wave as shown in (a). The output frequency of the waveform was $250kHz/64$ or $3.9kHz$. *Comparing FFTs of the two waveforms, they are very similar apart from the noise floor. FFT of the programmed waveform does not have any frequency component at clock frequency as there was no physical clock present in that measurement.*

in the row can be individually programmed to store a precise analog value. During the normal operation, a shift register scans through the entire row of programmed floating-gates and generates a sampled waveform at the output. In this architecture, W_1 to W_M can be any arbitrary set of waveforms that are programmed and can be used to modulate or demodulate any input signal. Details of the programming scheme such as speed and accuracy can be found in chapter [20] along with the Gate and Drain logic for programming. The generated waveforms are sampled in time–domain and can be cleaned by performing low–pass filtering to suppress the higher–order frequency components.

4.2.1 Frequency Performance

The frequency of the generated waveforms depend on the clock frequency and the number of floating–gate devices in each column. Frequency of the output waveform, f_{out} with N floating–gate devices is given by Mf_{clk}/N, where f_{clk} is the clock frequency and M is number of periods programmed in a particular row. Thus, the

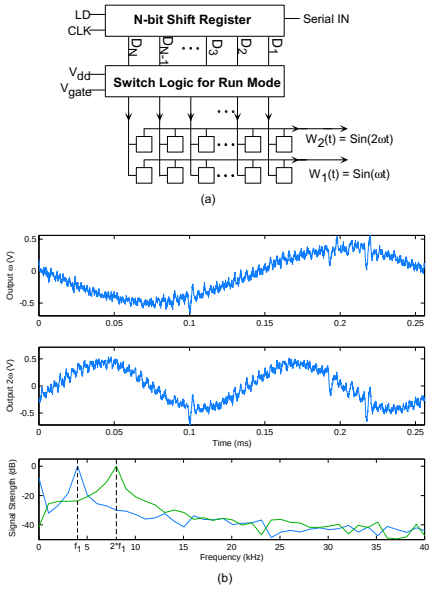


Figure 30. Generated output waveform at ω and 2ω.: Measurement showing the output waveforms when two rows (one cycle and two cycles) were programmed with a $100nA_{pp}$ sine wave riding on a $300nA$ DC current. The clock speed is $250kHz$ and the number of elements in a row are 64. The output signal frequency generated from the two rows is $3.9kHz$ and $7.8kHz$, respectively. As is clear, waveform generator can be used to generate arbitrary waveform with varying frequencies.

output frequency can either be increased by increasing the clock frequency, f_{clk} or by programming more than one period of the waveform on a single row of floating–gate devices as shown in Fig. 27 [40]. The latter will govern the LPF rejection requirements for getting clean output waveforms. Thus, the frequency of such a system is inherently limited to the frequency of clock that can be generated cleanly or the complexity of the LPF acceptable at the output. The shift register that scans through the row of floating-gate transistors during normal operation is designed for appropriate frequency performance and uses dynamic logic for fast response. Figure 26(a) shows the schematic of the D-flip flop (DFF) that is used to implement the shift register. The output of each DFF is buffered to drive the floating–gate capacitance. This analog implementation eliminates the need for an adder at the output as the addition of currents can be simply done by connecting the output of each floating–gate together. Each row has a cascode transistor for the reasons explained earlier. Apart from the

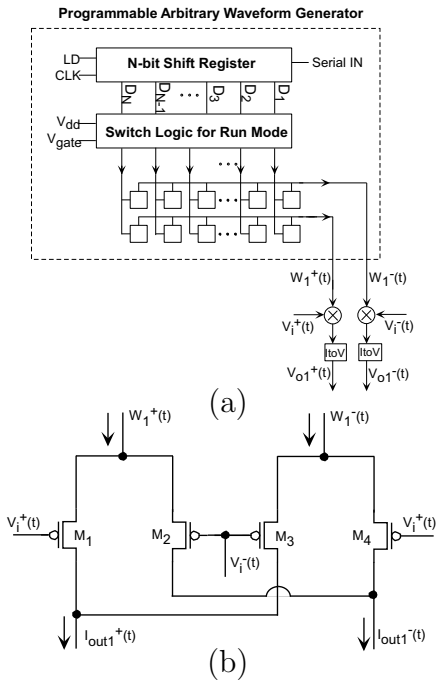


Figure 31. Block diagram showing modulation/demodulation.: (a) Block diagram for the analog modulator/demodulator system. It can be easily extended for multi-channel system by adding more rows to the waveform generator. (b) Circuit schematic for mixer implementation shown in (a).

speed of the shift register, the frequency response if also limited by the total line capacitance of each row. The output pole at the drains of the floating–gate row is given by,

$$P_{out} \approx \frac{g_{m,cas}}{C_{line}} \tag{47}$$

This frequency can be increased by programming all the floating–gates at a higher bias current and also by supplying a auxiliary bias current through the cascode all the time. The performance can also be improved a lot by using an active cascode structure. The performance will now depend on the input stage, which is a function of the number of parallel rows being driven and resistance of switches. High speed operation of the complete system puts a design constraint on the clock speed as well. The quality of clock in terms of rise-time, fall-time, and jitter along with coupling of the clock will affect the quality of the generated signal. Thus, generating a clean clock signal for high frequency applications becomes crucial.

4.2.2 Phase Noise and Quantization Error

The output of the waveform generator for a sinusoid can be expressed as

$$W_j = A_c \cos(\omega_c t + \phi_n(t)) \tag{48}$$

where ω_c is the desired output frequency and the phase $\phi_n(t)$ is referred to as the phase noise of the system. Figure 28 shows the spectrum of such a waveform when compared to an ideal output [40]. Phase noise is generated when the samples are randomly shifted off from the ideal output waveform. In the proposed system, any jitter in clock will be the major source of such noise. Thus, to generate a clean clock is a requirement to have better PAMD system performance. In Fig. 26(a), Clock and -Clock signals should be generated with equal delays such that there is no time difference at zero–crossing in order to minimize any phase noise generation. The distribution of Clock and -Clock to the entire shift register in the layout is critical for the performance of such a system. A tree layout, as shown in Fig. 26(b), was used with chain of inverter buffers to minimize any such delays. Along with these sources, any error in the programmed value of the floating–gate charge can also be modelled as phase noise in the output spectrum. Minimizing the error will further improve the phase noise performance. One other source of error in sampled systems in quantization error. The number of floating–gate devices in each row determine the quantization error in the output waveform. As in a DDS system, the quantization error appear as a periodic additive term rather than a random noise as long as the ratio of f_{clk} and f_{out} is a rational number [40]. Thus, the resulting error and its harmonics occur as spurs in the output spectrum. The amplitude of these spurs is determined by the programming accuracy of the floating–gate charge and can be suppressed by the LPF at the output.

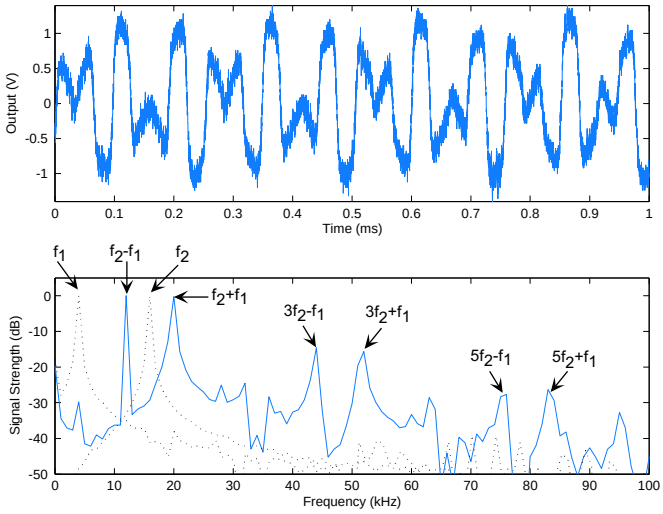


Figure 32. Measurement showing modulation.: Output waveform and spectrum when a $15.9KHz$ signal is modulated with a $3.9KHz$ signal.

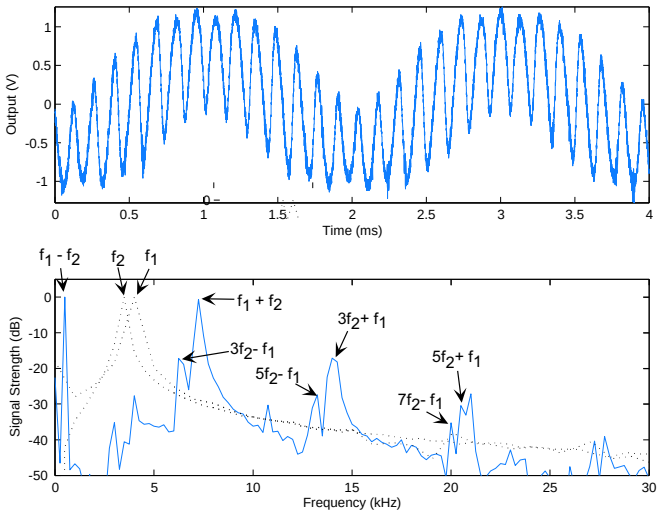


Figure 33. Measurement showing demodulation to near DC.: Output waveform and spectrum when a $3.4KHz$ signal is demodulated with a $3.9KHz$ signal. The output signal at $500Hz$ can be filtered to reject the high frequency components.

4.2.3 Measurement Results

In order to measure the performance of the waveform generator, simple current-amplifiers were used at the output along with I-V converters to measure the output reliably especially with lower current amplitudes. Figure 29(a) shows a measurement of a programmed $100nA_{pp}$ sine wave riding on a $300nA$ DC current. As evident from (1), programmed current shown in Fig. 29(a) is proportional to the charge stored on each floating-gate node. We obtained a worst case programming error of 0.2% and it takes about 10 pulses of $100us$ to programmed each floating-gate [20]. The FFT of this waveform is also shown and is clearly limited by the quantization noise. The FFT was performed assuming a $256us$ time-period for the entire programmed sine-wave. This was done in order to compare the results directly with the measured data when a clock of $250kHz$ is applied to the PAMD system. Figure 29(b) illustrates the output waveform as it looks when the clock of the shift register is turned ON. As can be seen from the FFT of the programmed charges and the output waveform, a clean frequency can be generated without any observable higher-order harmonics. The measurement is limited by the noise of the measurement setup. The FFT also shows the clock frequency and images of the signal around clock frequency. Thus, the system requires a clean clock signal and a programmable lowpass filter at the output to filter out anything outside the bandwidth of the desired output waveform. Figure 30 illustrates the measurement of the waveform generator block when programmed to ω and 2ω. Figure 30 shows that this waveform generator can clearly be used to synthesize any arbitrary waveform such as chirp or any other modulating waveform.

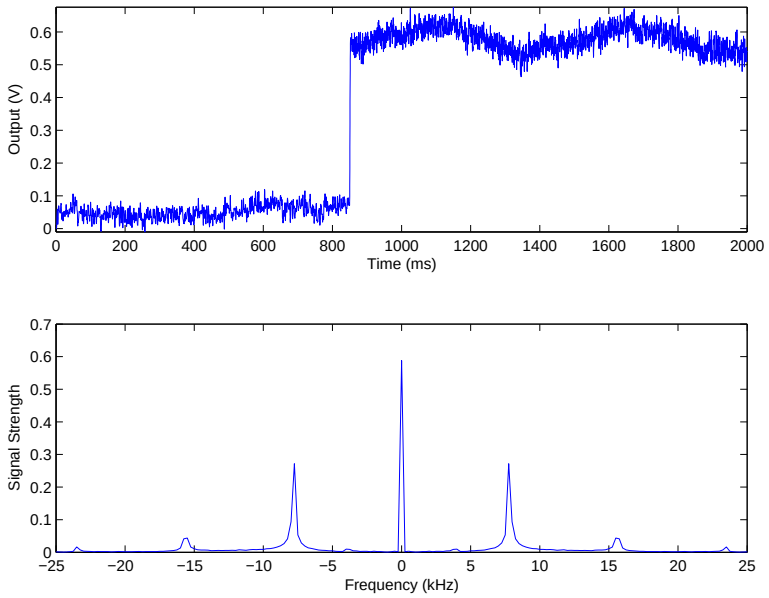


Figure 34. Measurement showing demodulation of input signal to DC.: Output waveform and spectrum when a $3.9KHz$ signal is demodulated with a $3.9KHz$ signal. The output signal at DC can be filtered to reject the high frequency component. This approach can be used to extract the spectral content of the input signal at desired frequencies. In the current experiment, the input signal was left running and output of the modulator was turned ON after some time to see the transition in the DC level of the output signal and was filtered to extract the low-frequency information. The output waveform still has a very slow AC component of approximately $1.5Hz$. This is attributed to the limited precision of the function generator used to provide the input signal.

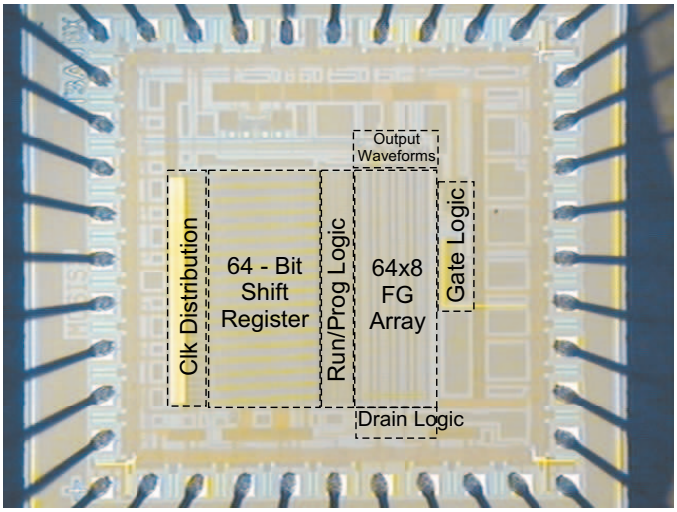


Figure 35. Die photograph of a 64x8 modulator system.: The PAMD IC was fabricated in 0.5μm MOSIS CMOS process and occupied an area of approximately 1mm^2. The fabricated IC can generate four fully–differential arbitrary analog waveform. The number of outputs can be easily increased without having to increase the area by a lot.

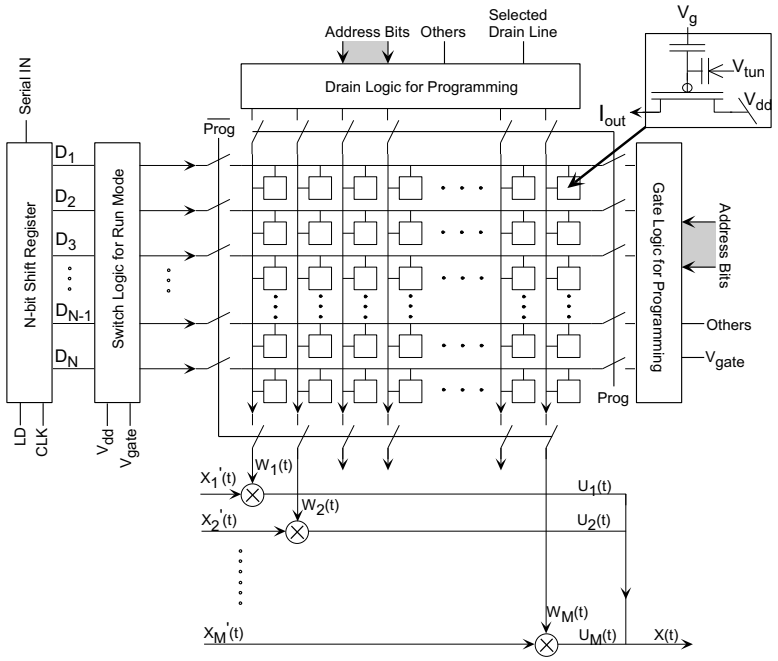


Figure 36. Floating-gate implementation of OFDM transmitter.: Analog implementation of OFDM transmitter using floating-gate devices. W_1 **to** W_M **can be sinusoidal waveforms (or any arbitrary set of waveforms) used to modulate the signal waveform. They can be programmed to give different waveforms whenever desired.**

4.3 Programmable Analog Modulator/Demodulator (PAMD) Architecture

Figure 31(a) shows the block diagram of the programmable analog modulator/ demodulator (PAMD) system using the floating-gate waveform generator. PAMD system has differential gilbert-cell mixers, shown in Fig. 31(b), at the output to modulate or demodulate the differential input signal. Figure 32 shows the output when a $15.9kHz$ input signal is modulated with the $3.9kHz$ signal generated by the modulator. The input signal, $15.9kHz$, is generated using a Stanford Research System (SRS) function generator. This input signal source has a limited phase noise performance. The $3.9kHz$ signal is generated with a sine wave programmed on a row of 64 floating-gates and using a clock speed of $250kHz$. Figure 32 illustrates the basic modulation operation and shows the FFT of the output spectrum. The output spectrum signal can be appropriately filtered to select the desired signal.

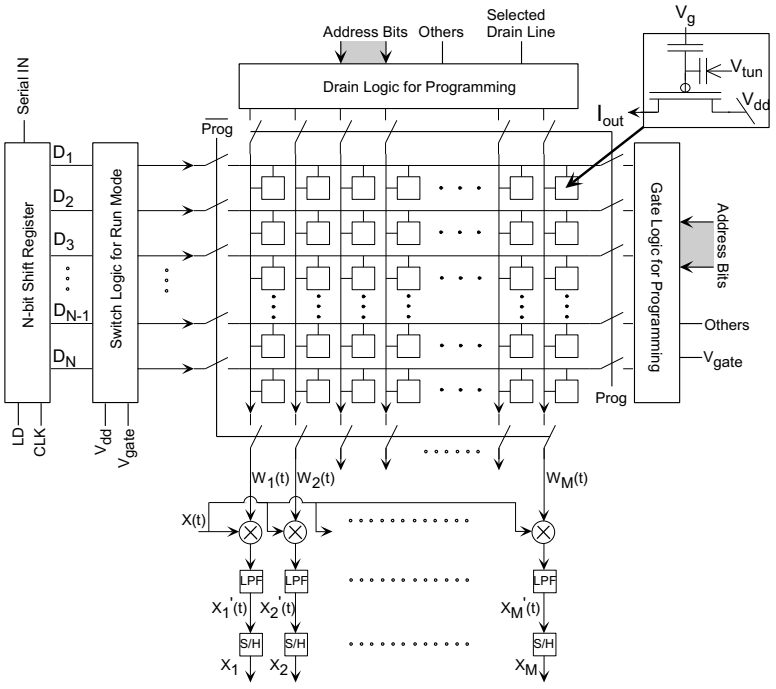


Figure 37. Floating-gate implementation of OFDM receiver.: Implementation of OFDM receiver using floating-gate devices. The columns can be programmed to similar waveforms as used for transmission. This can be used to bring the signal down to baseband.

Figures 33 and 34 show the demodulation operation to near DC and at DC for the input signal, respectively. Figure 33 shows the measurement when a $3.4KHz$ input signal is demodulated to $500Hz$ using the generated waveform, $3.9KHz$. This signal can be easily filtered from the spectrum to reject the high frequency spurious signal at $7.3KHz$. Figure 34 shows the demodulation to extract the DC signal strength of the input signal by demodulating it to DC. In the current experiment, the input signal was left running and output of the modulator was turned ON after some time to see the transition in the DC level of the output signal. The output waveform still has a very slow AC component of approximately $1.5Hz$. This is attributed to the limited precision of the function generator used to provide the input signal. As is clearly evident, this can be used to extract the spectral content of an input signal at desired frequencies by demodulating them with the desired frequencies and filtering the DC signal out. Figure 35 shows the die photograph of the PAMD IC that was fabricated

and tested. The total area occupied by the system was approximately 1mm^2. The fabricated IC can generate four fully–differential arbitrary analog waveform. The number of outputs can be easily increased without having to increase the area by a lot, as discussed earlier.

4.4 Applications

The proposed architecture can be used for a variety of other applications along with the described modulator/demodulator. The key advantage for the presented architecture is due to the ability to generate programmable analog arbitrary waveforms. One such application can be generating arbitrary waveforms to perform on-chip testing of other mixed–signal circuits and systems. The advantage of using such an approach is that it does not require multiple input analog pins for various test nodes inside the circuit. It only requires one digital clock input and is fully programmable depending upon the circuit under test. This can be easily made as part of a Built-in Self Test with a control loop to test various designs.

The presented PAMD architecture can also be used for a wide variety of communication schemes as mentioned in previous section. Figure 36 shows a possible implementation of the a transmitter using PAMD that can be used for various communication schemes such as Orthogonal Frequency Division Multiplexing (OFDM), where multiples of fundamental frequency are used to orthogonally modulate different channels. Each column of floating–gates can be programmed to generate a desired waveform as output, as shown in Fig. 30. Traditionally, these operations are performed as FFT/IFFT for OFDM in digital domain that are computationally area and power intensive [35]. These columns can also be programmed to generate arbitrary waveforms, which can find its application in many other areas such as generating chirp waveforms to perform modulation. This system can be used as the receiver also with little modifications, as shown in Fig. 37. The data converters required in

these implementations are operating on the baseband signal, thus making the design relatively simple.

The presented waveform generator can be used as a part of an adaptive equalizer system. It can be programmed to generate any waveform that can be used to perform equalization. The compact nature of the architecture and low power consumption makes it suitable for multiple-channel processing and array signal processing.

4.5 Summary

In this chapter, we presented an analog modulator/demodulator that can be used for various communication schemes and array signal processing applications. The proposed implementation enables a lot of signal processing in the analog domain before the signal is converted into digital domain and eliminates the need for expensive, wideband data converters. This approach can be both power and area efficient compared to existing implementations using DSPs for portable applications [34, 41]. PAMD consists of a programmable arbitrary waveform generator using floating-gate MOS devices. We presented results for the programmable waveform generator along with the spectral energy plot. We showed results with the basic operation of a modulator and demodulator operation. We also discussed and presented how to extract spectral content of an incoming signal at specific frequencies by performing auto-correlation using the proposed structure. The presented structure with proper design can be used for a variety of other applications as discussed and is being explored along those lines.

CHAPTER 5

CONTINUOUS-TIME OTA-C FILTERS

Continuous-time filters are another fundamental component of any analog signal processing system. The demand is high for analog filters with better performance in terms of speed and power consumption for systems with limited power budget. In this Chapter, we try to investigate the possible implementations of continuous time filters using floating–gate devices and their performance that will further give insight into implementation of programmable analog systems for analog signal processing. Continuous time filters, particularly $G_m - C$ filters, are the most often used solution for signal frequencies of several MHz [42] as problems such as jitter and high dynamic power make discrete-time filters impractical at such frequencies. These filters though have issues such as offsets due to device mismatch, limited linearity, and require additional circuity to tune the filter to get the desired response after fabrication [43, 44].

To address these issues, we show two kinds of $G_m - C$ filter implementations in this chapter and the next chapter. In this chapter, we discuss more of a traditional operational transconductance amplifier (OTA-C) based filter design. We present an approach that will not only help tune the filter to get desired Q and time constants but also compensates for any offset due to mismatch. In addition, the filter can be designed to have certain minimum linearity based on the capacitive attenuation at the input. These abilities will make these $G_m - C$ filters attractive to use even at lower frequencies [45]. We will first discuss design of fully–differential programmable operational transconductance amplifiers using floating–gate MOSFETs (FG-OTA) that are used as building block for the OTA-C filter realizations. We designed two different FG-OTA implementations with different common–mode feedback (CMFB) circuits. We will present the measured experimental results for the two FG-OTA structures and compare their performance. We also present experimental results for

the lowpass and bandpass second-order sections that were fabricated using floating–gate operational transconductance amplifiers (FG-OTAs).

Any higher order filter can be realized as a cascade of biquad filters. Although there are several ways to realize higher order filters, cascade filters are the easiest to design as well as to tune. We also present simulation results for higher-order filters using the presented second–order sections as building blocks.

5.1 Programmable Operational Transconductance Amplifiers (OTA)

The most important component in designing traditional $G_m - C$ filters is the design of OTAs. The transconductance (G_m) of these OTAs is the parameter that determines the frequency response and gain of these filters. Thus to have programmable filters, we essentially need programmable OTAs [46]. Traditional approaches to realize programmable OTAs include digital and master–slave tuning as well as schemes based on Multiple Input Floating-Gate (MIFG) transistors [47, 48]. Digital schemes used in filters are complex and consume silicon real estate. The MIFG approach has a lower area overhead; however, it fails to fully exploit the benefits of floating-gates – especially the ability to program them.

We present a true programmable approach to using single input floating-gate transistors in programmable OTA blocks and OTA-C filters. The circuit schematic of the floating-gate OTA (FG-OTA) with a novel floating-gate CMFB (FG-CMFB) circuit is shown in Fig. 38(a). This OTA will be referred to hereafter as FG-OTA1. FG transistors are used for the tail current source, differential input pair and in the output stage to implement the CMFB circuit. The programming of these floating–gates sets the bias currents for the OTAs (and hence the transconductance) as well as corrects for differential pair mismatch in OTAs and gradients across the die [49]. Figure 39(a), shows the second FG-OTA with conventional CMFB circuit. This OTA

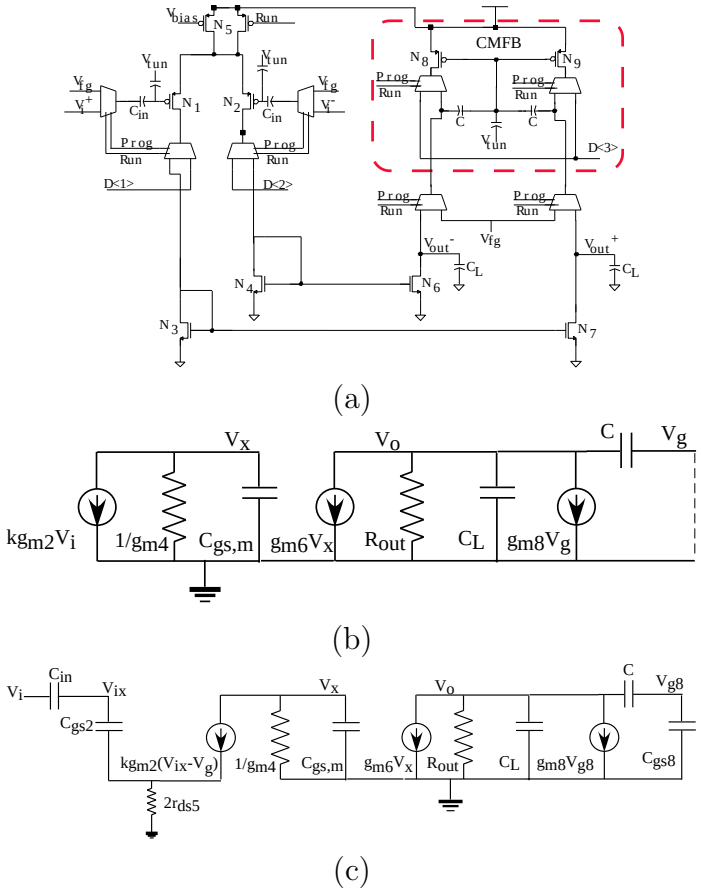


Figure 38. Fully differential FG-OTA with floating-gate CMFB circuit.(FG-OTA1): (a) Circuit schematic for the floating–gate operational transconductance amplifier (FG-OTA1) using a CMFB built into the OTA structure. The floating–gate capacitors around the output PMOS current source transistors form the CMFB circuit for FG-OTA1. (b) Small-circuit model for differential-mode analysis. (b) Small-signal model for common-mode analysis.

will be referred to hereafter as FG-OTA2. The corresponding CMFB circuit is shown in Fig. 39(b) that feeds back the error signal to the tail current of the OTA to correct for any common–mode variation. We will now discuss the basic operation and design of the two programmable OTAs and follow that by their experimental results. We will conclude this section by comparing the performance of the FG-OTA1 and FG-OTA2.

As will be discussed that while FG-OTA1 is compact and consumes less power, FG-OTA2 has the advantage of a higher common-mode feedback loop gain, better current mirror matching, higher output impedance with output cascoding and higher

differential open-loop gain. We will discuss the qualitative and quantitative analysis for both FG-OTA1 and FG-OTA2 to get an intuitive and analytical understanding of both the implementations. We follow this discussion with simulation and experimental results for both implementations. In the end, we summarize the comparison between the two structures in terms of design and performance.

5.1.1 Differential FG-OTA1

5.1.1.1 Basic Operation

Figure 38(a) shows the complete circuit schematic for the differential FG-OTA1. The biggest advantage of FG-OTA1 comes from the fact that the common–mode feedback (CMFB) is integrated in the OTA structure rather than having some additional circuitry. This makes the circuit compact and suitable for higher–order filter implementation where a number of OTAs are required. This implementation also helps in reducing the noise due the fact that there are no additional transistors added to the circuit.

Floating–gate transistors at the input are used to remove the input–referred offset, as discussed in Chapter 2. Input–referred voltage offset causes the drain currents, I_{d1} and I_{d2}, to be different for the same common–mode input. Programming the drain currents to be identical helps remove the offsets to a first–order approximation for the given conditions. If δI_d is the difference in the two currents due to the offsets, it gives a output voltage as

$$Vout^+ - Vout^- = \frac{g_{m6}}{g_{m4}} \delta I_d R_{out} \tag{49}$$

This voltage output can be considerable depending on the offset and can even saturate the amplifier when used in open–loop. Along with that, it limits the linear range of the amplifier along with the limitation on the minimum input level that can be detected without error. The input capacitor, C_{in}, also help in improving the linearity of the FG-OTA1 due to capacitive attenuation. The transconductance, G_m, of FG-OTA1 depends on the tail current and can be tuned by programming the tail current

floating–gate transistor.

Let us now qualitatively discuss the operation of the CMFB for FG-OTA1. The output PMOS current sources are implemented as floating–gate transistors. The advantage of this ie two-fold: 1) It helps remove any mismatch in the output current sources. 2) The floating–gate capacitors that are used for programming can be used in feedback to implement the CMFB. If currents through N_8 and N_9 are programmed to be identical to the current flowing through N_6 and N_7, common–mode of V_{out} will be biased in the middle of the rail. The basic operation of the CMFB circuit can be understood easily. If the output common–mode $V_{out,cm}$ increases for some reason, the voltage at floating–gate of PMOS current source increases due to capacitive coupling. The increase in floating–gate voltage decreases the drain current through the output PMOS sources and thus, brings down the $V_{out,cm}$.

The primary advantage for using the floating–gate (FG) capacitors in feedback is that they do not affect the DC gain by loading the output node, yet they perform the CMFB operation all the way down to DC. One would think that FG capacitors would load the amplifier at high frequencies and thus, degrading the frequency response. In reality, applications like $G_m - C$ filter implementation have external capacitors at the output node that determine the corners of the filter. Therefore, the FG capacitor can be lumped as part of the output capacitor and the size of the physical external capacitor can be adjusted accordingly. However in a lot of applications, the external capacitor is very large compared to the FG capacitor making their effect on the frequency response negligible.

5.1.1.2 Quantitative Analysis

We now describe the differential and common-mode analysis of the FG-OTA1 in Fig. 38 to understand the operation of the circuit in order to design for performance. Simple expressions for the transfer function is derived. These are used to gain intuitive understanding of the FG-OTA as well as do the first-pass hand design. Figure 38(b)

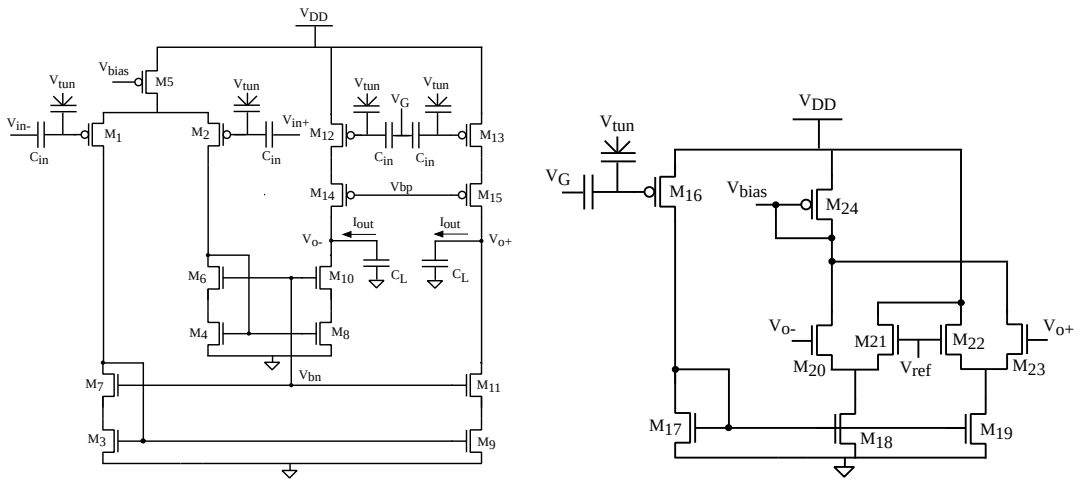


Figure 39. Programmable floating-gate OTA with common mode feedback: (a) Circuit schematic for the programmable floating-gate OTA (FG-OTA). Inherent offsets of the amplifier are compensated by programming the floating-gate transistors. Floating-gate transistors M_1 and M_2 are used to eliminated the input referred offset of the amplifier. Transistors M_{12} and M_{13} account for any error at the output. (b) Circuit schematic showing the common mode feedback circuit (CMFB) used for the programmable FG-OTA. Transistor M_{16} sets the bias current for the FG-OTA. Hence, the G_m of this amplifier can be adjusted by programming M_{16}.

shows the small signal differential-mode half-circuit. Since the idea is to get an intuitive understanding of the circuit, we will neglect the effect of the overlap capacitance, C_{gd}. This capacitance in reality will give rise to a zero that can be ignored to simplify the analysis. Writing KCL at nodes V_x and V_o we have,

$$\frac{V_o}{V_x} = \frac{g_{m6}R_{out}}{(1+sC'_LR_{out})} \tag{50}$$

$$\frac{V_x}{V_i} = \frac{g_{m2}}{g_{m4}\left(1+\frac{sC_{gs,m}}{g_{m4}}\right)} \tag{51}$$

where,

$$C'_L = C_L + C,\;\; C_{gs,m} = C_{gs,4} + C_{gs,6} + C_{db,2},\;\; R_{out} = r_{ds,6} \parallel r_{ds,8} \tag{52}$$

Hence,

$$A_{dm} = \frac{V_o}{V_i} = \frac{g_{m6}g_{m2}R_{out}}{g_{m4}\left(1+\frac{sC_{gs,mirror}}{g_{m4}}\right)(1+sC'_LR_{out})} \tag{53}$$

Assuming the current–mirror ratio between transistors N_6 and N_4 to be n, we can simplify the DC differential voltage gain to be,

$$A_{dm,DC} = n g_{m2} R_{out} \tag{54}$$

Figure 38(b) shows the small signal common-mode half-circuit. To simplify the analysis we assume $g_m >> g_{ds}$. This results in the following expression for the common-mode transfer function.

$$\begin{aligned} \frac{V_o}{V_{cm}} &= \frac{g_{m6}}{g_{m8}\kappa_{out} + s\left(\frac{C_L + C_{gs8}}{C + C_{gs8}}\right)} \cdot \left(\frac{1}{sC_{gs,m} + g_{m4}}\right) \\ &\quad \cdot \frac{g_{m2}}{\left(1 + 2\kappa_{in}\left(g_{m2} - sC_{gs2}\right) r_{ds5}\right)} \end{aligned} \tag{55}$$

where,

$$\kappa_{in} = \frac{C_{in}}{(C_{in} + C_{gs2})}, \;\; \kappa_{out} = \frac{C}{(C + C_{gs8})}, \;\; C_{gs,m} = C_{gs,4} + C_{gs,6}$$

To get an intuitive understanding of the transfer function (55) is split up into the DC gain and poles as below where p_1 is the dominant pole assuming the current–mirror ratio to be n as discussed above.

$$A_{cm,DC} = \frac{n}{2 r_{ds5} \kappa_{in} g_{m8} \kappa_{out}} \tag{56}$$

$$p_1 = \frac{g_{m8}\kappa_{out}}{C_L + \kappa_{out} C_{gs}} \approx \frac{g_{m8}\kappa_{out}}{C_L} \tag{57}$$

$$p_2 = \frac{1 + 2\kappa_{in} r_{ds5} g_{m2}}{2\kappa_{in} r_{ds5} C_{gs2}} \tag{58}$$

$$p_3 = \frac{-g_{m4}}{C_{gs,m}} \tag{59}$$

The CMRR for the FG-OTA1 circuit can be easily computed now using the $A_{dm,DC}$ and $A_{cm,DC}$.

$$CMRR = \frac{A_{dm,DC}}{A_{cm,DC}} = g_{m2} R_{out} 2 r_{ds5} \kappa_{in} g_{m8} \kappa_{out} \tag{60}$$

These equations help to get a basic understanding while designing the FG-OTA1 by hand and further help in understanding the simulation results while designing the

final circuit. The location of poles aid in design the FG-OTA1 for stable operation by keeping a good phase margin. We now will discuss the same analysis for FG-OTA2 before presenting the experimental results for both and concluding with a comparison in terms of design and performance.

5.1.2 Differential FG-OTA2 design and analysis

5.1.2.1 Basic Operation

Figure 39(a) show the schematic of the differential floating-gate (FG) operational transconductance amplifier (FG-OTA2) [50] structure using a conventional CMFB circuit. The FGs at the input, M_1 and M_2, can be programmed to correct for any input offsets and improve the input linear range as in the case of FG-OTA1. Output floating-gate transistors, M_{12} and M_{13}, help correct any mismatch in the output current-source transistors, thereby aiding common-mode feedback circuit (CMFB) in improving the CMRR. The output stage of the FG-OTA2 was cascoded to give a high output resistance, which decreases the dominant pole of the OTA-C block, giving it a more ideal integrator behavior over a wider frequency range. The high output resistance also gives higher gain for FG-OTA2. The cascoded NMOS current mirrors reduce the channel length modulation effect when mirroring currents.

Figure 39(b) shows the CMFB circuit for the differential FG-OTA2. Any common–mode variation in $V_{out,cm}$ is compared with V_{ref} through this differential amplifier. For example, if $V_{out,cm}$ is increases due to some reason. This would increases the current in transistor M_{24} and thus, in the tail current–source M_5. Hence, the current in output NMOS current sinks, M_8 and M_9, increases bringing down the common–mode $V_{out,cm}$. The bias current and, hence, the corner frequency of FG-OTA2 is determined by the current flowing through the floating-gate transistor M_{16}. Thus, the G_m of FG-OTA2 can be adjusted by programming M_{16} similar to FG-OTA1. The output common–mode in FG-OTA2 is externally set by V_{ref} and can be fixed to any desired voltage. The CMFB circuit can be designed for desired common–mode gain while

not affecting core OTA structure.

5.1.2.2 Quantitative Analysis

We will discuss the equations governing the differential and common-mode gain for the FG-OTA2 as we did for FG-OTA1. Using the small-signal model half-circuits for FG-OTA2 (as was done for FG-OTA1) the DC gain equations can be obtained as

$$A_{dm,DC} = g_{m1}\frac{g_{m9}}{g_{m3}}R_{out} \tag{61}$$

$$\begin{aligned} A_{cm,DC} &\approx \frac{g_{m24}}{2r_{ds5}g_{m5}g_{m20}} && (62)\\ CMRR &\approx g_{m1}g_{m20}2r_{ds5}R_{out} && (63)\\ Rout &= g_{m11}r_{ds11}r_{ds9}//g_{m15}r_{ds15}r_{ds13} && (64) \end{aligned}$$

The poles for differential and common–mode gain expressions can also be obtained by analyzing the small–signal circuits. The expressions for the poles are,

$$P_{1,dm} \approx \frac{1}{C_L R_{out}} \tag{65}$$

$$\frac{P_{2,dm} \approx g_{m3}}{C_{gs,mirror}} \tag{66}$$

$$\frac{P_{1,cm} \approx g_{m20}}{C_L} \tag{67}$$

$$\frac{P_{2,cm} \approx g_{m24}}{(C_{gs,24} + C_{gs,5})} \tag{68}$$

The above two FG-OTA implementations were fabricated in a standard CMOS technology. We measured the performance of these fabricated circuits and compared them with the simulation results. We will now discuss the experimental results for both the structures before comparing the two implementations.

5.1.3 Simulation and Experimental Results

Simulation results are presented for FG-OTA1 and FG-OTA2. Figure 40(a) shows the simulated differential and common-mode frequency response for the FG-OTA1. Simulated small-signal frequency response of FG-OTA2 is shown in Fig. 40(b). From

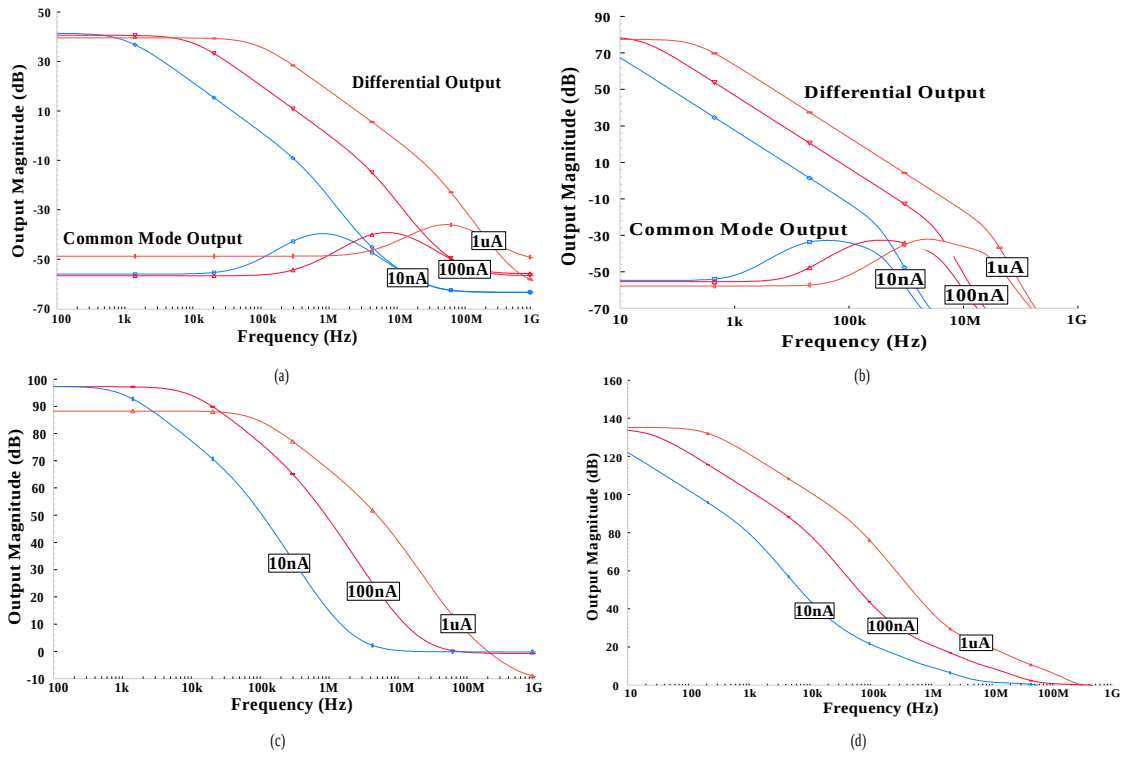


Figure 40. Simulation results for the programmable FG-OTAs: SPICE simulation results of small signal common-mode and differential-mode response of (a) FG-OTA1. (b) FG-OTA2. Plot shows data for three values of OTA bias currents–10nA, 100nA and 1μA. SPICE simulation results of CMRR versus frequency of (c) FG-OTA1 (d) FG-OTA2. Plot shows data for three values of OTA bias currents–10nA, 100nA and 1μA. Experimental results of FG-OTA1 can be seen in Fig. 41.

the figure, it can be seen that the -3dB frequency is directly related to the bias current. An order of magnitude increase(decrease) in the bias current, corresponds to an analogous increase(decrease) in the corner frequency. The CMRR for both OTAs was also simulated as a function of frequency. Figures 40(c) and 40(d) show the corresponding results. The simulated CMRR for FG-OTA1 and FG-OTA2 was 90dB and 140dB, respectively.

The IC prototype was fabricated in a 0.5μm CMOS MOSIS process. The prototype includes both FG-OTA1 and FG-OTA2. DC and AC responses of the FG-OTAs were measured. Figure 41(a),(b) shows the measured transient common-mode response of the two OTAs. For both circuits, a DC input common mode sweep was performed to determine the input common-mode range. Figure 41(c) shows results

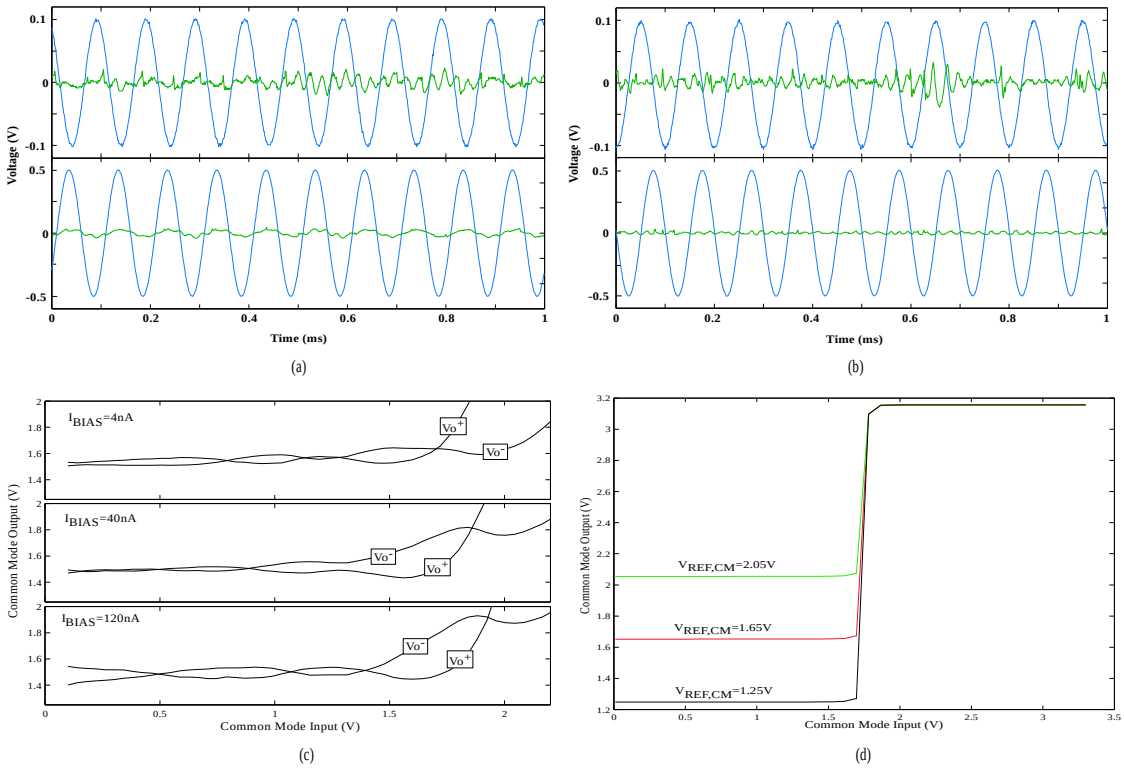


Figure 41. Common-mode measurement for the programmable FG-OTAs: Transient common-mode response (a) FG-OTA1 circuit (b) FG-OTA2. Response is shown for 10kHz input common-mode signal at 200mVpp and 1Vpp. The input signal rides on a DC level (not shown) of V_{CM}= 1.2V. Input common-mode DC sweep response for FG-OTA1 are shown in (c). Plots show output common-mode voltage for three values of bias currents–4nA, 40nA and 120nA. Output common-mode voltage is held at 1.55V. Plot (d) shows output common-mode voltage for FG-OTA2 as the reference voltage to the CMFB circuit is varied.

for FG-OTA1, while Fig. 41(d) shows the corresponding curves for FG-OTA2. It is seen that for both OTAs the input common-mode range to restricted to less than 1.7V. This limitation is caused by bias transistor operating out of saturation region due to voltage headroom issues.

Multiple copies of these OTAs were realized with different input capacitance C_{in} between the input node and the floating-gate node. The objective, was to see the potential increase in linearity due to capacitive attenuation as is shown later. For FG-OTA1, the differential inputs were swept to obtain the curves in Fig. 42(a). From these curves the DC gains were computed and they were close to the theoretical values.

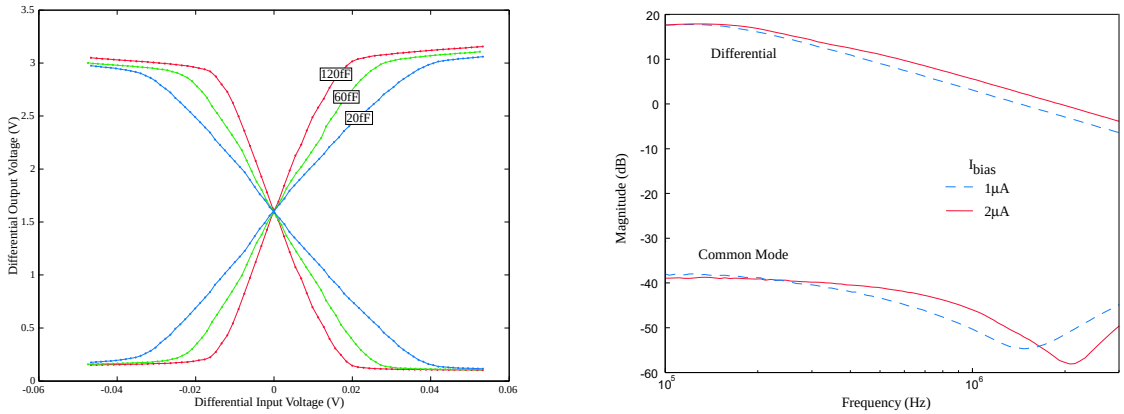


Figure 42. Differential gain measurement and frequency response for the programmable OTA: (a) DC differential input sweep for the FG-OTA1 circuit with varying C_{in} values– 20fF, 60fF and 120fF. Measured DC gains are 40.01V/V, 60.77V/V and 95.75V/V, respectively. The gain is a function of the capacitance C_{in} connecting the differential input to the floating-gate node. (b) Experimental frequency response of FG-OTA1 for two different programmed bias currents.

As can be clearly seen, the input linear range increases as the C_{in} is decreased. This is due to fact that the capacitive attenuation of the input signal increases as the input capacitance decreases.

Experimental frequency response measurements of FG-OTA1 are shown in Fig. 42(b). It may be noted that the differential gain is lower than predicted by SPICE. This can be attributed partly to test setup inaccuracies and to the difficulty involved in measuring the open-loop gain of a high gain amplifier.

5.1.4 Comparison of FG-OTA1 and FG-OTA2

In the previous sections, we presented two implementation of programmable OTAs using floating–gate devices along with their results. In this section, we will summarize the performance of the two implementations along with elaborating on the advantages and disadvantages of the two implementations.

The primary advantage of FG-OTA1 as compared to FG-OTA2 was that it did not required any external circuit for common–mode feedback and thus, making it compact. It uses the same floating–gate capacitors, which were used to match the

output current sources, in feedback to obtain the CMFB. These capacitors do not affect the DC gain by loading the output node at DC, yet they perform the CMFB operation all the way down to DC. The capacitors can be made a part of the physical capacitor at the output of the OTA that is used to implement the filter corners.

Any mismatch in the PMOS current sources and NMOS current mirrors will cause the common–mode voltage to move accordingly such that the currents are balanced. This can be considered an advantage or disadvantage of the FG-OTA1 implementation. The advantage is that it can be used to set the common–mode voltage where desired by accurate programming, depending on the next stage. This eliminates the need for a dedicated reference to generate the common–mode voltage as is the case for FG-OTA2. Even though the common–mode output voltage for both V_{out}^{+} and V_{out}^{-} moves, it will move together up or down to balance the currents as long as transistors N_6 and N_7 are matched reasonably. The matching between N_6 and N_7 can become a real issue for this implementation as any mismatch may cause the two common–mode output voltages to move differently and may even saturate one side while keeping the other balanced. This effect can be sorted out by careful programming of either the input transistors or output current sources to account for this mismatch as was done for our measurements. However, this process is iterative and can be very slow as far as real operation is concerned. This limits the use of this configuration as such in filter implementations. This effect becomes more prominent if the output stage is cascoded to increase the DC gain.

Although FG-OTA2 has the disadvantage of consuming more area and requiring a dedicated reference to set the common–mode, it has several advantages that make it relatively easy to use in filter designs. FG-OTA2 can be easily cascoded to give a high output resistance, which decreases the dominant pole of the OTA-C block, giving it a more ideal integrator behavior over a wider frequency range. The high output resistance also results in higher gain for FG-OTA2. The cascoded NMOS

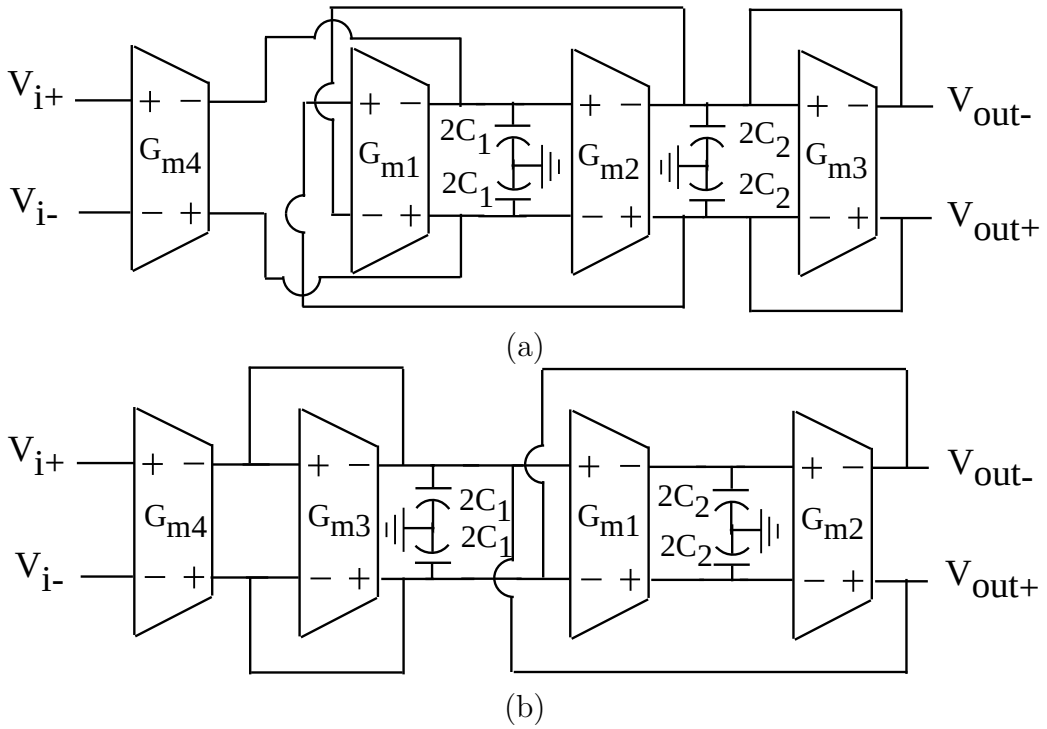


Figure 43. Programmable, fully differential $G_m - C$ second-order sections: (a) Block diagram of a standard $G_m - C$ Lowpass biquad. (b) Block diagram of a standard $G_m - C$ Bandpass biquad.

current mirrors reduce the channel length modulation effect when mirroring currents. The output common–mode in FG-OTA2 is externally set by V_{ref} and can be fixed to any desired voltage and does not depend a whole lot on the device properties unlike FG-OTA1. This helps in the cascading these OTAs to design higher–order filters without worrying about the common–mode of the next stage.

Due to the advantages of FG-OTA2 as compared to FG-OTA1 in terms of ease of design and performance, FG-OTA2 was used to design second–order programmable filter sections. In the next section, we discuss the design of these second–order sections using FG-OTA2s and present their experimental results.

5.2 Programmable $G_m - C$ Filter Sections

As discussed earlier, continuous-time can easily operate on high–speed signals and have a significant speed advantage over their switched–capacitor counterparts. One

of the major challenge in designing these filters though is to have some tuning circuitry that can accurately set the filter response after fabrication [43, 44]. This is due to the fact that their filter coefficients are determined by capacitors and transconductance values.

We present an approach using our programmable OTAs that will not only help tune the filter to get desired Q and time constants but also compensates for any offset due to mismatch. We discuss the design of programmable second–order sections because any higher order filter can be realized as a cascade of several second–order biquad sections. A biquad structure is a second–order filter structure that allows for independent tuning of the center frequency and quality factor, Q, based on the circuit components. It can be easily modified to be used as a lowpass, bandpass or highpass filter configurations. The general biquad transfer function is given as,

$$\frac{V_{out}}{V_{in}} = \frac{k_2 s^2 + k_1 s + k_0}{s^2 + \frac{s\omega_0}{Q} + \omega_0^2} \tag{69}$$

Here, ω_0 and Q are the pole frequency and pole Q, respectively, whereas k_0, k_1 and k_2 are arbitrary coefficients that determine the filter type. Although there are several ways to realize higher order filters, cascade filters are the easiest to design as well as to tune.

5.2.1 Second–Order Sections (SOSs)

We used the FG-OTA2, as discussed in the previous section, to design OTA-C based second–order biquad sections. We designed and fabricated both a programmable, fully differential lowpass (LPSOS) and a bandpass second–order section (BPSOS) on a 0.5μm n-well CMOS process available through MOSIS. Any higher order filter can be realized as a cascade of biquad filters. Although there are several ways to realize higher order filters, cascade filters are the easiest to design as well as to tune. The paper presents experimental results from two such programmable biquads: the lowpass second-order section (LPSOS) and the band pass second-order section (BPSOS) as

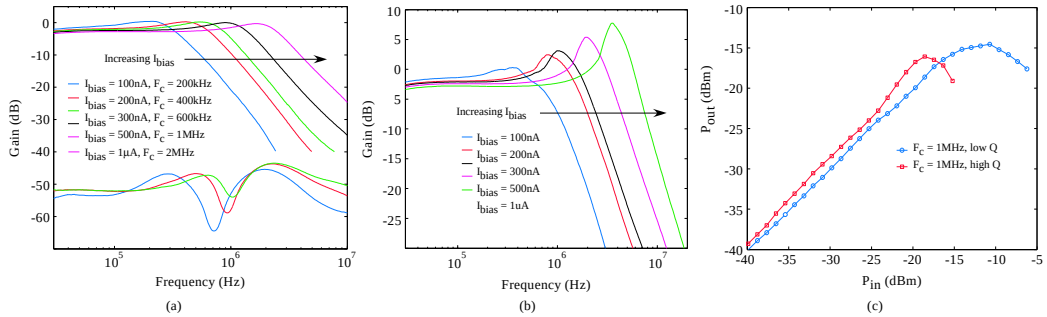


Figure 44. Lowpass SOS Experimental Results: (a) Measured differential and common-mode gain for the LPF programmed to different corner frequencies ($200kHz$ - $2MHz$). The measured common mode gain for lowpass biquad agreed with simulated values. (b) Measured differential gain for the LPF showing the Q variation for different programmed bias currents. (c) Measured plot to compute the 1-dB compression point for a LPF tuned at 1MHz for two different programmed Q values. The currents were initially programmed to give a flat response and then current setting the lower time constant was increased using injection to make the poles complex and give a Q-peak.

shown in Figure 43(a) and 43(b). These basic building blocks can be used to design higher order bandpass filters for analog signal processing applications. FG-OTAs are used as programmable G_m elements in Figure 43(a) and 43(b).

5.2.1.1 Lowpass SOS

Figure 1(a) shows the block diagram of the lowpass biquad (LPSOS) using FG-OTA's. The transfer function of the SOS is:

$$\frac{V_{out}}{V_{in}} = \frac{\frac{G_{m4}}{G_{m1}}}{\frac{s^2 C_1 C_2}{G_{m1} G_{m2}} + \frac{s G_{m3} C_1}{G_{m1} G_{m2}} + 1} \tag{70}$$

If $C = C_1 = C_2$ and $G_m = G_{m1} = G_{m2}$, the time constant (or corner frequency) and Q for complex-conjugate poles is given by:

$$\tau = \frac{1}{\omega} = \frac{C}{G_m}, Q = \frac{G_m}{G_{m3}} \tag{71}$$

A desired corner frequency can be obtained by programming the bias current that control G_m, while the Q of the filter can be independently set by adjusting G_{m3}.

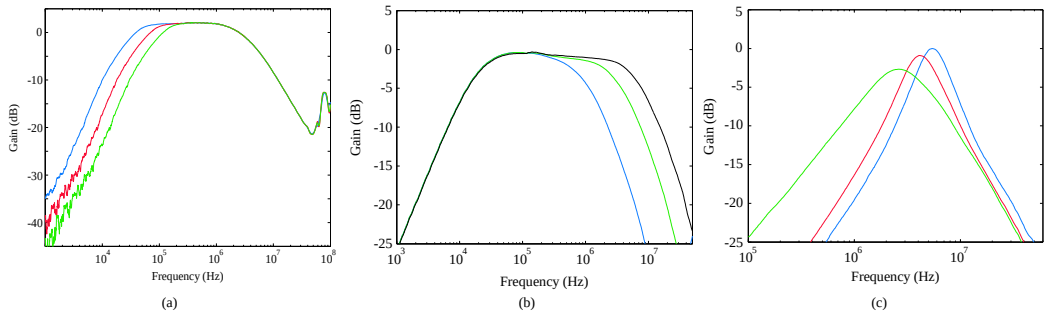


Figure 45. Bandpass SOS Experimental Results: (a) Experimental results showing the programming of the low corner of the Bandpass filter. Corner frequencies were programmed at $25kHz$, $50kHz$ and $100kHz$. (b) Experimental results showing the programming of the high corner of the Bandpass filter. Corner frequencies were programmed at $1MHz$, $2MHz$ and $4MHz$. (c) Experimental results showing programming of the low corner of the Bandpass filter for different Q values. As the G_m is increased, Q increases and the center frequency also increases as predicted by (5).

5.2.1.2 Bandpass SOS

Figure 1(b) shows the block diagram of a $G_m - C$ BPSOS using four FG-OTAs. The transfer function of the SOS is give by :

$$\frac{V_{out}}{V_{in}} = \frac{\frac{sG_{m4}C_1}{G_{m1}G_{m2}}}{\frac{s^2C_1C_2}{G_{m1}G_{m2}} + \frac{sG_{m3}C_1}{G_{m1}G_{m2}} + 1} \tag{72}$$

If $C = C_1 = C_2$ and $G_m = G_{m1} = G_{m2}$, the time constants for real poles, using the dominant pole approximation, are given by:

$$\tau_l \approx \frac{G_{m3}C}{G_m^2}, \tau_h \approx \frac{C}{G_{m3}} \tag{73}$$

The time constant (or corner frequency) and Q for complex conjugate poles is given by:

$$\tau = \frac{1}{\omega} = \frac{C}{G_m}, Q = \frac{G_m}{G_{m3}} \tag{74}$$

It can be observed from (4) and (5) that the corners and the center frequency of the BPSOS can also be set by programming the FG-OTAs.

5.2.2 Experimental Results

Figure 44(a) shows measured data of the differential gain of the LPSOS for different programmed G_m's while keeping the ratio G_m over G_{m3} constant. As can be seen, the

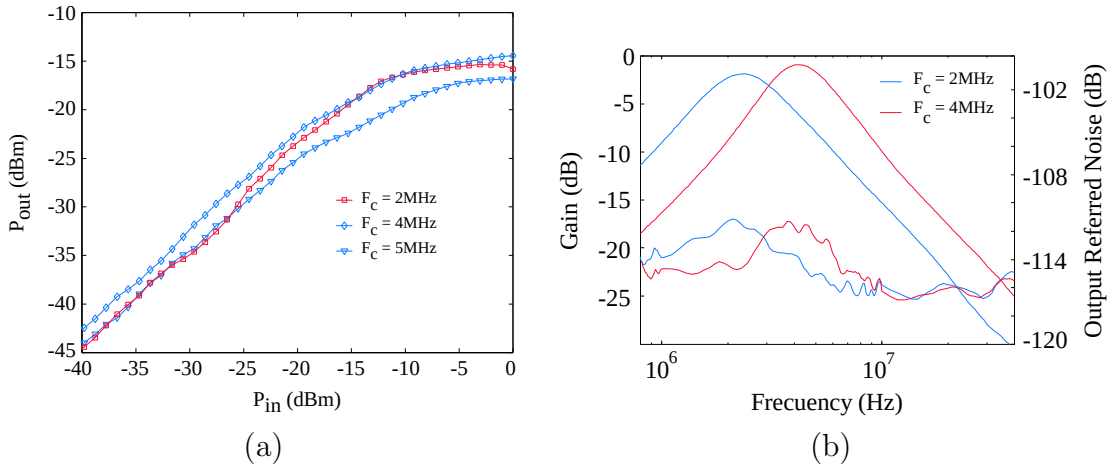


Figure 46. BPSOS Performance: ((a) 1-dB compression points for a BPF tuned at different frequencies. (b) Output referred spot noise of Bandpass filter tuned at 2 MHz and 4 MHz. The noise obtained at these frequencies is mostly thermal.

corner frequencies move linearly ($200kHz$ - $2MHz$) with the bias current as long as the input transistors operate in sub-threshold. This is due to the fact that transconductance varies linearly with bias current in the sub-threshold region. Figure 44(a) also shows the common-mode gain for these structures for different bias currents suggesting a good CMRR. The experimental results correlated well with the simulations for these plots.

Experimental results of the LPSOS for different programmed Q values are shown in Figure 44(b). This was done by programming different G_m values. The corner frequency also moves as expected from (2). The Q values can be independently adjusted by programming G_{m3}.

Figure 44(c) shows the measured output power for varying input power of the lowpass SOS when tuned to 1MHz corner for the two different Q values. This measurement can be used to find the 1-dB compression point of the system by doing a simple curve fit. The linearity of the system deteriorates with higher Q due to higher gain in the system. The measured 1-dB compression for the high Q and low Q case was 160 mV_{pp} and 280 mV_{pp}, respectively.

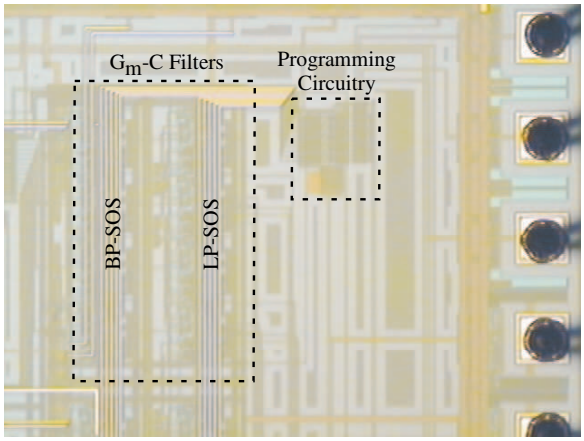


Figure 47. Die Micrograph: The circuit prototype was fabricated in a 0.5μm n-well CMOS process. The total area for the BPSOS and LPSOS is 0.135mm^2.

Figure 45(a) shows the experimental response of the BPSOS with different programmed G_m's. As predicted in (4) the low corner changes while keeping the high corner constant (G_{m3} is kept fixed). Figure 45(b) shows the measured response for the BPSOS, where the high corner has been moved independent of the low corner frequency. It follows from (4) that this is accomplished by programming the bias currents controlling G_{m3}, and keeping the ratio G_{m3} over ${G_m}^2$ constant. Figure 45(c) shows the filter response for different Q values. Here G_m was programmed so complex poles were obtained. The center frequency will also vary as a function of G_m. Careful programming of these FG-OTAs can give varying values of Q for different center frequencies.

The measurement used to compute 1-dB compression of the BPSOS for three different corner frequencies, with similar Q and gain, is shown in Figure 46(a). The linearity is similar for the three different frequencies in this case by design and is about 397 mVpp (or -11 dBm). Figure 46(b) shows the output-referred noise spectrum of the programmed BPSOS with center frequencies of 2 MHz and 4 MHz. The spectrum looks like that of the tuned filter response as expected. The noise at these frequencies

is purely thermal as can be observed from the measured data. The worst-case input-referred spot noise power occurs at the center frequencies and is -109 dBm. Figure 47 shows the circuit prototype fabricated in a 0.5μm n-well CMOS process. The total area for the BPSOS and LPSOS is $0.135mm^2$. The filters can be programmed to desired corner frequencies and Q values.

5.3 Summary

In this chapter, we discussed design of programmable continuous time filters. These filters can be programmed to operate anywhere from audio band to lower MHz band after fabrication. We presented experimental results from two programmable $G_m - C$ biquads: the lowpass second–order section and the band pass biquad. Any higher order filter can be realized as a cascade of biquad filters. Although there are several ways to realize higher order filters, cascade filters are the easiest to design as well as to tune. Based on the above measurements, it is possible to design a fully programmable higher order bandpass filter that can be tuned to different responses (like Butterworth, Chebyshev) at different frequencies by programming appropriate coefficients.

CHAPTER 6

CONTINUOUS-TIME C^4-FILTERS

As mentioned in the previous chapter, with increasing trend of designing power-efficient analog circuits for portable applications, the demand is high for $G_m - C$ filters with better performance in terms of speed, area and power consumption. Traditional $G_m - C$ filter implementations based on Operational Transconductance Amplifiers (OTAs) that were discussed in previous chapter are area-intensive, thus making them unsuitable for filter-bank applications.

In this chapter, we present a programmable, continuous-time, bandpass filter section that is compact and power efficient. This programmable filter element, shown in Fig. 48, will be referred to as the capacitively coupled current conveyer (C^4) due to the similarity to the current conveyer structure [Ismail REF]. The corner frequencies and Q-values for this element depends on the bias currents. These current sources are implemented using floating-gate devices and can be accurately programmed [20]. Earlier discussions have showed an initial approach and potential applications, especially in the audio band, for these filters [51, 11]. In this chapter, we will present a rigorous design procedure for the filter section.

We also present the design of higher-order filters using our programmable filter element, as is depicted by the cascade of our filter sections in Fig. 48. The design equations presented can be easily used to synthesize first-pass circuit parameters, according to the desired specifications, using any standard software, such as MATLAB. The designed higher-order filters can easily be tuned to desired transfer functions, such as Butterworth and Chebyshev, after fabrication by simply programming floating-gate current sources.

In section 6.1, we discuss the design of the programmable 2^{nd}-order bandpass element. We present all the design equations to be used in obtaining parameter

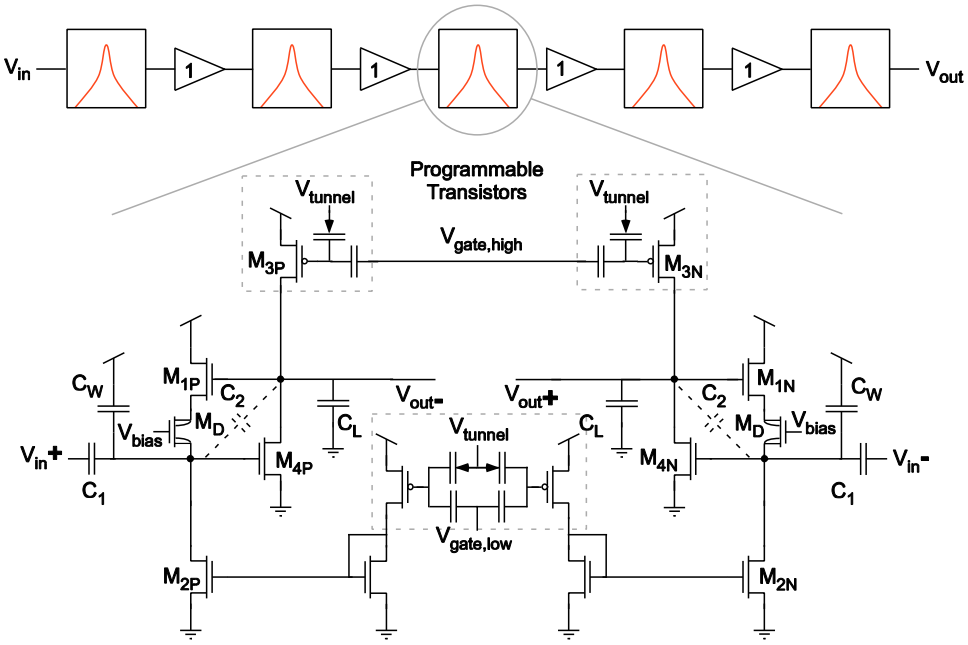


Figure 48. Block diagram and schematic of the filter element.: Block diagram of 10^{th}-order filter and circuit schematic of the core filter element. Floating–gate transistors can be programmed to set the desired bias current and, thus, accurate time constants and quality factor, Q. All other parameters can also be set using capacitor ratios. Transistor M_D, as shown by a special symbol, is a short–channel device that can be used to increase linearity at the low corner frequency.

values for first-hand design. Section 6.2 presents an equivalent model for high Q cases. This simplified model can be used to determine various performance parameters. In section 6.3, we present the measured results for 2^{nd}- and 4^{th}-order filters. Section 6.4 presents the design of a 6^{th}- and 10^{th}-order filter using our core programmable filter section. We also present measured results for the designed filters programmed with a Butterworth approximation. We conclude the discussion in section 6.5 with a summary of performance.

6.1 Design Considerations of Programmable Bandpass C^4 Element

Our core programmable 2^{nd}-order filter element was developed from the autozeroing floating-gate amplifier (AFGA) [52], shown in Fig. 49(b). The lower time-constant in the AFGA is small since it is set by hot-electron injection current and tunneling

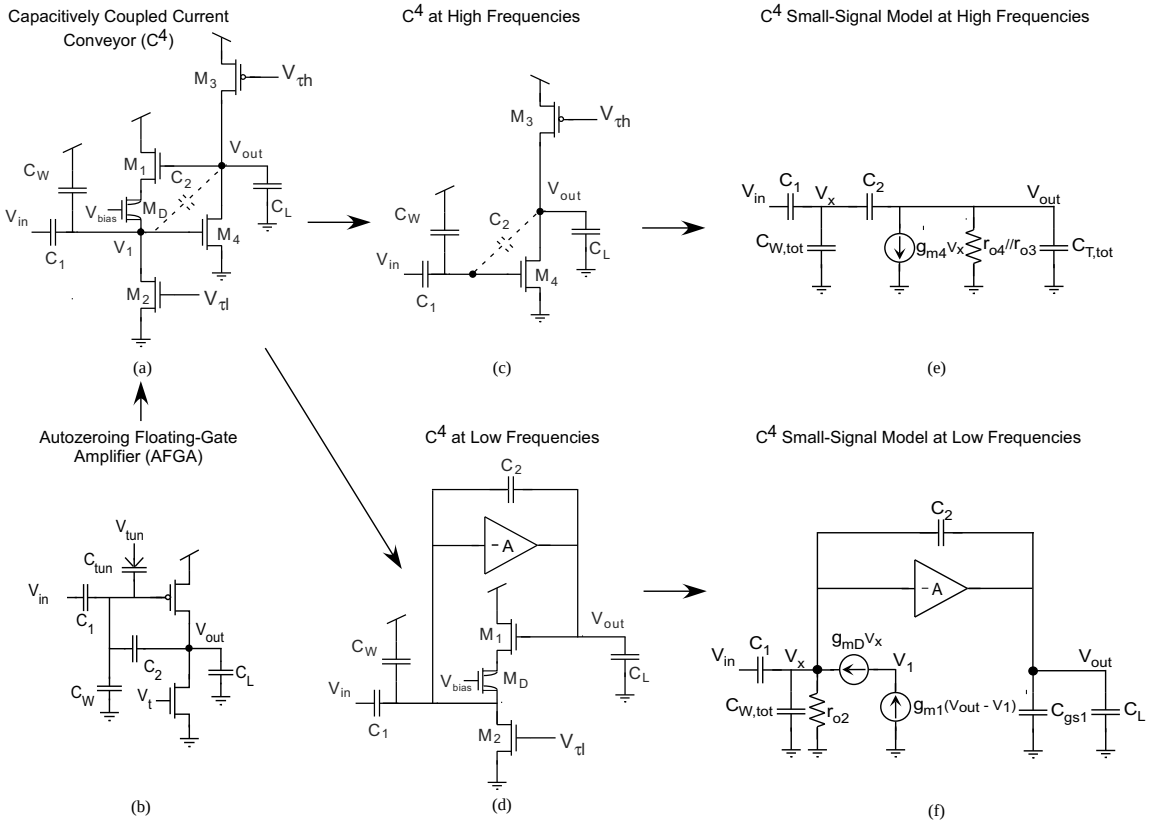


Figure 49. Evolution of C^4 filter section along with the equivalent circuits.:(a) Circuit schematic of Autozeroing Floating-Gate Amplifier (AFGA) [52]. The lower time-constant is set by constant tunneling and injection. (b) Schematic of capacitively-coupled current conveyor (C^4). The time-constants is set by transconductance of transistors and can be well-controlled (see (1), (2)). (c) Equivalent circuit schematic of C^4 at high frequencies. The feedback loop consisting of M_1 is non-functional at these frequencies. (d) Equivalent circuit schematic of C^4 at low frequencies. The common-source amplifier with transistor M_4 acts as a constant gain (A) amplifier in the feedback. (e) Small-signal model for the high-frequency equivalent circuit. This model can be used to compute linearity at the high-frequency corner. (f) Small-signal model for the low-frequency equivalent circuit and can be used to compute linearity at the low-frequency corner.

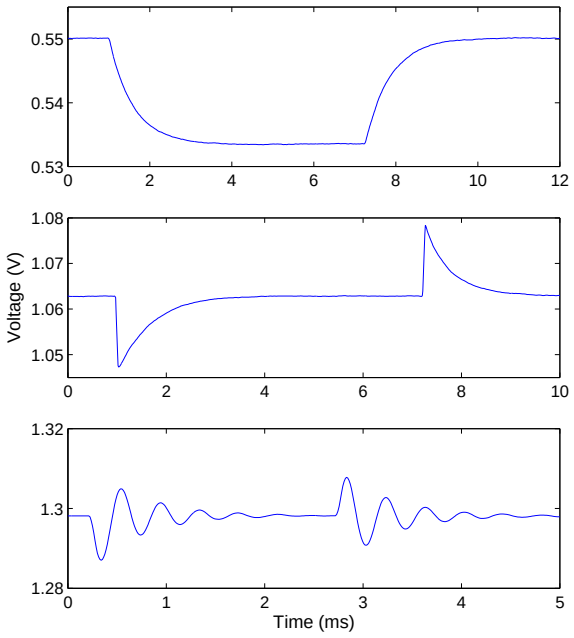


Figure 50. Step response of the C^4.: (Top) Step response of the C^4 when biased as an integrator. (Middle) Step response of the C^4 when biased as a differentiator. (Bottom) Step response of the C^4 when the two corners have crossed each other, thus given slight resonance within the filter response.

current, which are both typically small. This circuit was modified such that both time-constants can be set using transistor bias currents, and the resulting circuit, shown in Fig. 49(a), is a simplified half circuit of the C^4 [53]. By adding programmability through floating-gate transistors, the complete C^4 is as shown in Fig. 48. Transistor M_D, as shown by a special symbol, is a short–channel device that can be used to increase linearity at the low corner frequency. This will be illustrated later in the discussion.

Using the simplified half-circuit, shown in Fig. 49(a), of the programmable core filter element, the transfer function can be obtained by analyzing the small-signal circuit model for this half circuit. The transfer function is given by

$$\frac{V_{out}}{V_{in}} = -\frac{C_1}{C_2}\frac{\frac{sC_2}{g_{m1}}(1-\frac{sC_2}{g_{m4}})}{s^2\frac{(C_TC_o-C_2^2)}{g_{m1}g_{m4}} + s(\frac{C_2}{g_{m1}} + \frac{C_2}{g_{m4}}(\frac{C_o}{C_2}-1)) + 1} \tag{75}$$

where the low and high time constants (τ_l and τ_h), and the high-frequency zero (τ_f)

can be observed as

$$\tau_l = \frac{C_2}{g_{m1}}, \qquad \tau_h = \frac{(C_T C_o - C_2^2)}{C_2 g_{m4}}, \qquad \tau_f = \frac{C_2}{g_{m4}}$$

The low and high time constants are set independently of each other by programming g_{m1} and g_{m2} (shown in Fig. 49(a)), respectively. As a result, the C^4 can be programmed to act purely as an integrator or a differentiator, as is shown in Fig. 50. By moving the time constants closer to each other, the C^4 takes on a bandpass response. Crossing the time constants even introduces slight resonance into the filter response, as is also shown in Fig. 50. The zero τ_f is designed to be at sufficiently high frequency such that it lies well outside the passband and does not affect the response of the filter section. The quality factor, Q, and the center frequency for a particular value of bias currents are given by:

$$Q = \frac{\sqrt{(C_T C_o - C_2^2) g_{m1} g_{m4}}}{C_2 g_{m4} + C_L g_{m1}}, \qquad \tau = \sqrt{\frac{C_T C_o - C_2^2}{g_{m1} g_{m4}}}$$

For high values of Q (> 1), the expression for Q and τ can be reduced to

$$Q = \frac{\sqrt{C_T C_o g_{m1} g_{m4}}}{C_2 g_{m4} + C_L g_{m1}}, \qquad \tau = \sqrt{\frac{C_T C_o}{g_{m1} g_{m4}}}$$

The total capacitance, C_T, and the output capacitance, C_o, are defined as $C_T = C_1 + C_2 + C_W$ and $C_o = C_2 + C_L$. Transconductances g_{m1} and g_{m4} depend on the current flowing through transistor M_1 and M_4, respectively. The gain of the filter element is set by capacitor ratios and can, thus, be set accurately. The value of Q can be programmed by changing the ratio g_{m1}/g_{m4}. Figure 51 shows the plot of Q versus I_{d1}/I_{d4}. The plot clearly illustrates that maximum Q peak occurs for a certain value of I_{d1}/I_{d4} (thus, g_{m1}/g_{m4}) and goes down as the ratio is either increased or decreased. This is as predicted by 76. Assuming capacitance C_2 is small in comparison to C_L, the maximum Q-peak value, as shown in Fig. 51, can be derived from 76 to be

$$Q_{max} \approx \frac{1}{2}\sqrt{\frac{C_T}{C_2}} \tag{76}$$

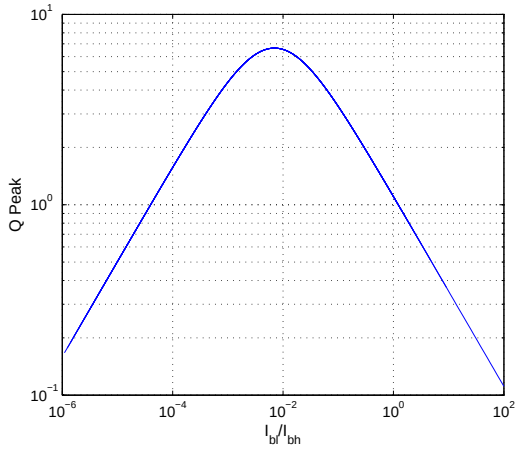


Figure 51. Q peak versus bias current ratio.: A maximum Q peak is defined for a given ratio of bias currents. As the current ratio changes, the Q peak value decreases.

The value of Q_{max} that can be obtained from a designed C^4 can be increased when the drawn capacitance C_2 is made small in comparison to C_L. In the case of no drawn physical capacitance C_2, the effective capacitance depends on C_{gd4} and C_{gs1}. The short-channel device, M_D as shown in Fig. 48, helps in alleviating the effect of C_{gs1} on the value of Q by reducing the effect of this coupling. Depending on the center frequency and Q requirements, these equations can be used to compute initial (W/L)s for each transistor depending upon the bias current and transistor region of operation. We now will derive and present equations assuming the transistors are operating in sub-threshold, which is usually the case for frequencies up to low MHz. These equations can be easily extended to the above-threshold region.

The initial (W/L)s can be used to compute the parasitic capacitances, provided the bias voltages at each node is given. These parasitic capacitance values will give the values of drawn physical capacitances that will also affect the performance parameters. The drain current, I_d, for a nMOS transistor in sub-threshold is given by

$$I_d = I_o e^{(\kappa V_g - V_s)/U_T} e^{V_d/V_A} \tag{77}$$

where I_o is a process dependent constant. V_g, V_s, V_d, V_A and U_T are the gate, source, drain, Early, and thermal voltage, respectively. Applying KCL at nodes V_{out} and V_1

in Fig. 49(a) and neglecting the Early affect, we get

$$I_o e^{\kappa V_1/U_T} = I_o e^{(V_{DD}-\kappa V_{\tau h})/U_T} \tag{78}$$
$$I_o e^{(\kappa V_{out}-V_1)/V_{A,D}} = I_o e^{\kappa V_{\tau l}/U_T}$$

where $V_{A,D}$ is the early voltage for the short-channel device, M_D, and I_o for PMOS and NMOS were assumed to be same. Solving these two equations gives the node voltages as,

$$V_1 = (\frac{V_{DD}}{\kappa} - V_{\tau h}) \tag{79}$$
$$\kappa V_{out} = V_1 + \frac{\kappa V_{A,D} V_{\tau l}}{U_T}$$

The total capacitance that will affect the final filter response is given by

$$C_{W,tot} \approx C_{W,drawn} + C_{gs,4} + C_{db,2} + C_{sb,D} \tag{80}$$
$$C_{2,tot} \approx C_{2,drawn} + C_{gd,4}$$
$$C_{L,tot} \approx C_{L,drawn} + C_{db,4} + C_{db,3} + C_{sb,D}$$

Using the above design equations, the circuit parameters meeting the desired specifications for the first-hand design can be easily synthesized using any software such as MATLAB. As can be seen from the above equations, the corner frequency and the Q-value also depend on the transconductances and, therefore, the DC bias current. Thus, the filter element can be easily fine-tuned after fabrication to the desired corner frequencies and Q-values by programming the g_{m1} and g_{m4}. This programmability is achieved using the floating-gate current sources [20], as shown in Figure 48. These floating-gate transistors (M_{3N}, M_{3P}, M_{5P} and M_{5N}) can be accurately programmed to any desired current level, as will be discussed in section III. Transistors M_{1N}, M_{1P}, M_{4P} and M_{4N} can operate in sub-threshold or in moderate inversion depending on the desired frequency response.

Figure 49(d) shows the equivalent circuit schematic of C^4 for low-frequency operation. Transistors M_3 and M_4 form a high-gain inverting amplifier at low frequency

corner. The short-channel device, M_D as shown in Fig. 48, helps to increase linearity at the lower corner frequency by source degeneration. To derive the linearity, using KCL at node V_{out} and V_1 gives

$$V_1 = (1 + \frac{C_L}{C_2})V_{out} \tag{81}$$
$$C_T \frac{dV_1}{dt} = C_2 \frac{dV_{out}}{dt} + C_1 \frac{dV_{in}}{dt} + I_{\tau,l}(e^{\frac{\kappa V_{out}}{V_{A,D}}} - 1)$$

Neglecting the transient current through the transistor M_1 as compared to the capacitive currents at the corner frequency, input linearity can be obtained as

$$V_{Li} = \frac{V_{A,D}}{\kappa} \frac{C_T C_o}{C_2 C_1} (1 - \frac{C_2^2}{C_T C_o}) \tag{82}$$

The increase in linearity can be derived from the small-signal model of the circuit shown in Fig. 49(f) and is given approximately by $g_m r_o$. This increase is $V_{A,D}/U_T$ for the subthreshold operation of transistors M_1 and M_D. Transistors M_1 and M_D usually operate in subthreshold as the bias current required to set the lower time-constant is typically small. Figure 49(c) shows the equivalent circuit schematic of C^4 for high-frequency operation. The linearity at the higher corner can be set by capacitor, C_W, due to capacitive attenuation at the input. The input and output linearity at the high-frequency corner can be obtained using 77 along with Fig. 49(e). Assuming that the current through transistor M_4 will be small as compared to the capacitive currents above the high-frequency corner, the linearity for sub-threshold operation can be determined by,

$$V_{Li} = \frac{U_T}{\kappa} \frac{C_T}{C_1} (1 - \frac{C_2^2}{C_T C_o}) \tag{83}$$
$$V_{Lo} = \frac{U_T}{\kappa} \frac{C_T}{C_2} (1 - \frac{C_2^2}{C_T C_o})$$

The output referred noise of the 2^{nd}-order section tuned to a particular response is

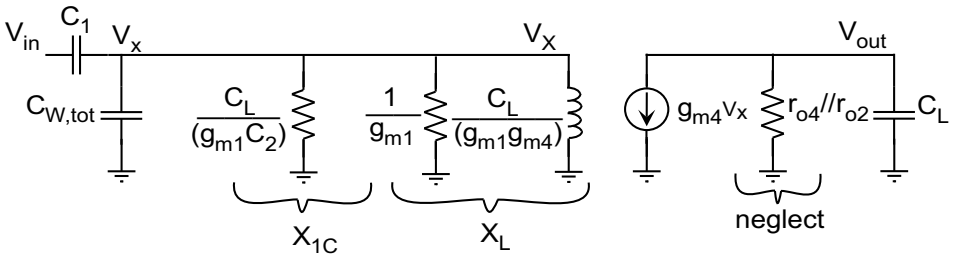


Figure 52. Model of C^4 for Q > 0.5.: Equivalent small-signal model of the C^4 developed for Q > 0.5 showing the effective inductance and capacitance that depends on the circuit parameters. This model can be used to get a intuitive feeling and hand calculate a lot of performance parameters for the high-Q case.

given by :

$$V_{no}^2 = (\frac{1}{s^2\tau_l\tau_h + s(\tau_l + \tau_f(\frac{C_o}{C_2} - 1)) + 1})^2[(\frac{sC_2 - g_{m4}}{g_{m1}g_{m4}})^2(I_1^2+I_2^2)+(\frac{sC_T + g_{m1}}{g_{m1}g_{m4}})^2(I_3^2+I_4^2)] \tag{84}$$

where I_1, I_2, I_3, I_4 are noise currents (thermal and flicker noise contribution) for transistors $M_{1N}, M_{1P}, M_{2N}, M_{2P}, M_{3N}, M_{3P}, M_{4N}$ and M_{4P}, respectively. As can be seen, the transfer function of the noise depends on the response of the filter and the circuit parameters. This expression can be used to design the filter element for good noise performance.

6.2 Equivalent Model for High Q case

To understand better the high Q (>0.5) case, we develop a small-signal model of C^4 that can be used to find the dependence of performance in terms of circuit parameters. Figure 49(a) shows the schematic of half-circuit for C^4. The voltage gain around the high-gain stage is given by,

$$\frac{V_o}{V_1} = \frac{-g_{m4}r_{04}}{1 + sC_Lr_{04}} \approx \frac{-g_{m4}}{sC_L} \tag{85}$$

If C_2 is small, then at high frequencies, using the Miller approximation in the small-signal model, the feed-forward current through C_2 can be neglected. The Miller

capacitance is given by

$$C_{Miller} = \left(1 + \frac{g_m 4}{sC_L}\right) C_2 \tag{86}$$

This gives the reactance at that node as,

$$X_{1C} \approx \frac{1}{s\left(1 + \frac{g_{m4}}{sC_L}\right) C_2} \approx \frac{1}{\frac{g_{m4}}{sC_L} C_2 s} \approx \frac{C_L}{g_{m4} C_2} \tag{87}$$

Figure shows the circuit schematic that can be used to compute the effective susceptance (X_L). Applying a test signal V_t and using KCL at the test node gives,

$$I_t + g_{m1}\left(V_o - V_t\right) = 0 \tag{88}$$

$$I_t = g_{m1}\left(V_t - V_o\right) \tag{89}$$

$$sC_L V_o + g_{m4} V_t = 0 \tag{90}$$

$$V_o = -\frac{g_{m4}}{sC_L} V_t \tag{91}$$

Solving for I_t using the above equations,

$$I_t = g_{m1}\left(1 + \frac{g_{m4}}{sC_L}\right) V_t \tag{92}$$

This gives the effective susceptance looking into the test node as

$$X_L = \frac{sC_L}{sC_L g_{m1} + g_{m1} g_{m4}} \tag{93}$$

Figure 52 shows the developed equivalent small-signal schematic for the Q > 0.5 case. This model can be used to evaluate the performance of the filter-section for high-Q cases. Using the small signal model shown in Fig. 52,

$$\frac{V_x}{\frac{sC_L}{g_{m1} g_{m4}}} + g_{m1} V_x + g_{m4} \frac{C_2}{C_L} V_x + s\left(C_1 + C_W\right) V_x = sC_1 V_{in} \tag{94}$$

$$\frac{V_x}{V_{in}} = \frac{sC_1}{\frac{g_{m1} g_{m4}}{sC_L} + \left(g_{m1} + g_{m4} \frac{C_2}{C_L}\right) + sC_T} \tag{95}$$

$$\frac{V_o}{V_x} = \frac{-g_{m4}}{sC_L} \tag{96}$$

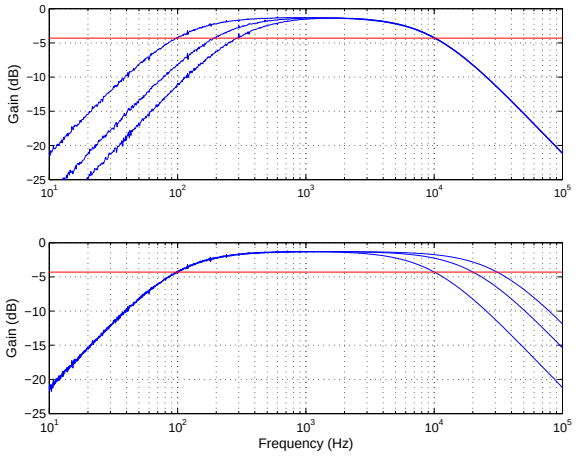


Figure 53. Measurement showing the programming of high and low corner frequencies.: The measured frequency response showing how the high and low corner frequencies can be programmed separately.

$$\frac{V_o}{V_x} = \frac{-sC_1 g_{m4}}{s^2 C_T C_L + s\left(g_{m1} C_L + g_{m4} C_2\right) + g_{m1} g_{m4}} \tag{97}$$

$$\approx \frac{-sC_1 g_{m4}}{s^2 C_T C_O + s\left(g_{m1} C_O + g_{m4} C_2\right) + g_{m1} g_{m4}} \tag{98}$$

This derived transfer function is similar to the original transfer function (given in 1) except the high frequency zero τ_f. This equivalence shows that this circuit model can be used to compute the performance of filter section for high Q (> 1).

6.3 Experimental Results for Bandpass Filter Sections

Based on the design equations discussed, we designed and fabricated 2^{nd}- and 4^{th}-order filter sections in 0.5μm CMOS technology. The designed filters had our floating-gate MOS transistors that can be programmed to have any desired bias current. Figure 53 plots the frequency response measurement of the 2^{nd}-order filter section. The plot clearly illustrates that both high (10kHz, 11kHz, 12kHz) and low (100Hz, 200Hz, 300Hz) corners can be individually programmed to desired frequency accurately using our floating–gate technology. Figure 54 shows the measured response of a 2^{nd}- and 4^{th}-order filter when programmed over decades of frequency. The 4^{th}-order filter

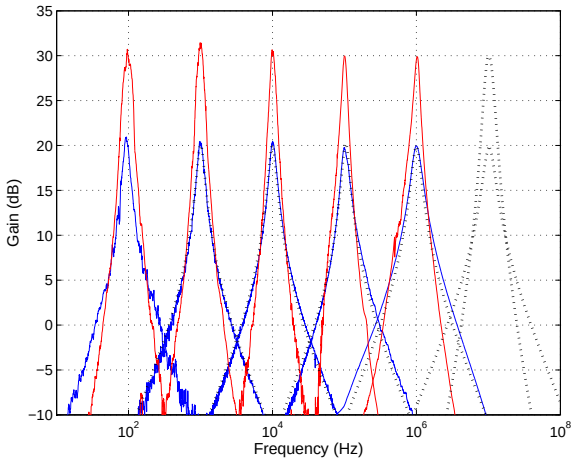


Figure 54. Measurement showing the programmed corner frequencies.: The measured frequency response showing that the filter can be programmed over a wide range of frequencies from 10Hz - 10MHz. Results are shown for 2^{nd}- and 4^{th}-order filters. Simulations of the filter, shown as dashed lines, matched well with the measured results.

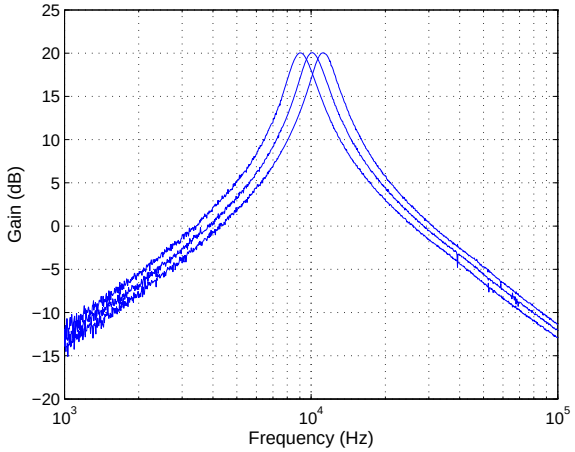


Figure 55. Measurement showing tuning of the filter element.: Measured frequency response of 2^{nd}-order filter tuned at 9KHz, 10KHz and 11KHz. Plot shows that the center frequencies can be fine-tuned by setting the desired bias current accurately using floating–gate transistors.

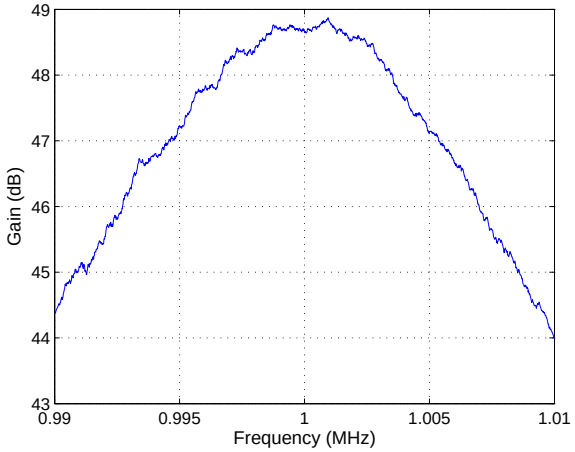


Figure 56. Q-tuning measurement: Measurement showing a Q-value of 70 obtained for a 4^{th}-order filter tuned at 1MHz center frequency. The value of Q can be determined by the 3-dB bandwidth and the center frequency.

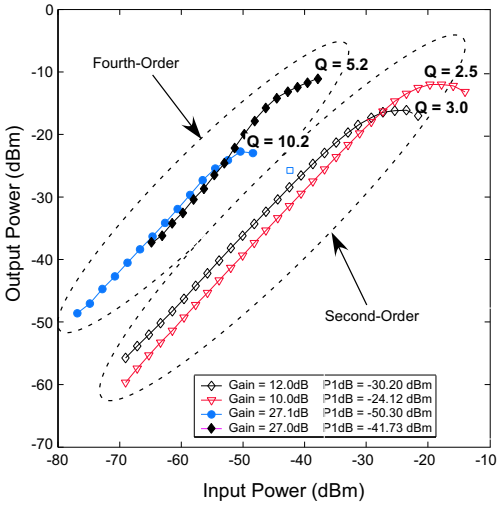


Figure 57. 1-dB compression measurement.: Measurement to compute 1-dB compression point for different values of Q for 2^{nd}- and 4^{th}-order sections.

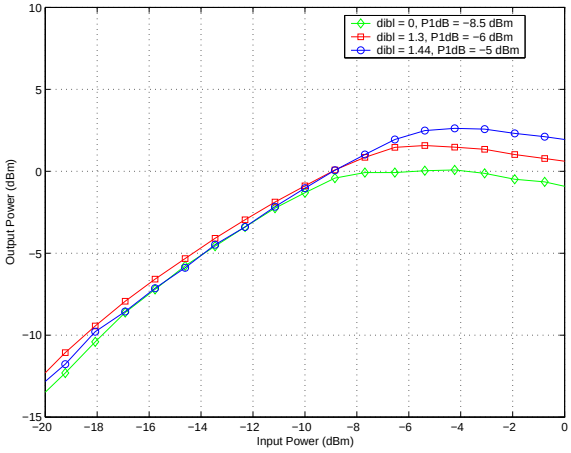


Figure 58. Effect of Vbias on linearity.: Measurement showing the improvement in 1-dB compression point as the bias voltage, Vbias, is increased. This increase in linearity is due to the source-degeneration effect.

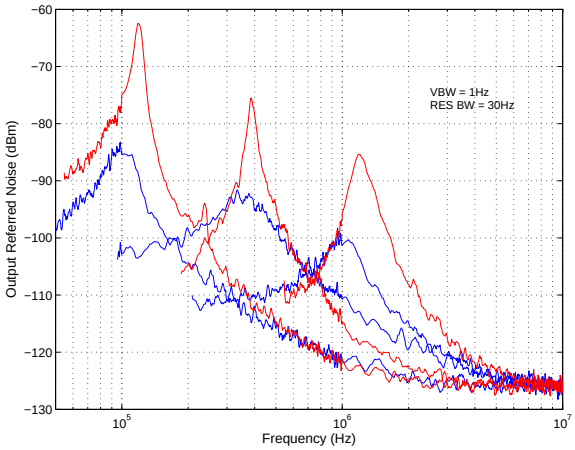


Figure 59. Noise measurements for 2^{nd}- and 4^{th}-order sections.: Plot showing the measured output-referred noise spectrums of 2^{nd}- and 4^{th}-order filters tuned at different frequencies.

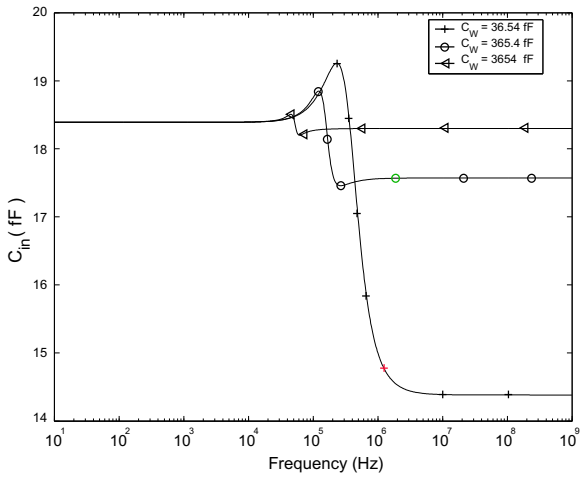


Figure 60. Input capacitance dependance on frequency.: Simulation results showing that the input capacitance of the C^4 varies with frequency.

was built by cascading two programmable 2^{nd}-order sections. The filter responses can be programmed anywhere from 100Hz to 10MHz. Simulations of the filter sections matched well with the measured response as can be seen from Fig 54. The measurements were limited to 1MHz due the output buffers ($f_{-3dB} = 10$MHz). Figure 55 shows the filter response (Q > 0.5) of a 2^{nd}-order section when fine-tuned over a small range of frequencies (9-11kHz). Figures 54 and 55 show that the filter topology can be both programmed over a wide frequency range and fine tuned over a small frequency range, if required. The designed 2^{nd}- and 4^{th}-order filter sections can be programmed to give Q-values up to 9 and 70, respectively. Figure 56 shows the measured plot of a 4^{th}-order filter tuned at 1MHz to have a Q of 70.

Figure 57 shows the measurement to compute the 1-dB compression point for 2^{nd}- and 4^{th}-order sections for two different values of programmed Qs. As expected, the linearity degrades as the Q-value increases. The values of linearity for the 2^{nd}- and 4^{th}-order sections tuned to have a Q of 2.5 and 5.2, respectively, at 1MHz were found to be -24dBm (83mV_{pp}) and -42dBm (11.5mV_{pp}), respectively. Figure 58 shows the measurement to compute the 1-dB compression point for different values of V_{bias} for a 2^{nd}-order section programmed to have a low Q. It can be clearly seen that the

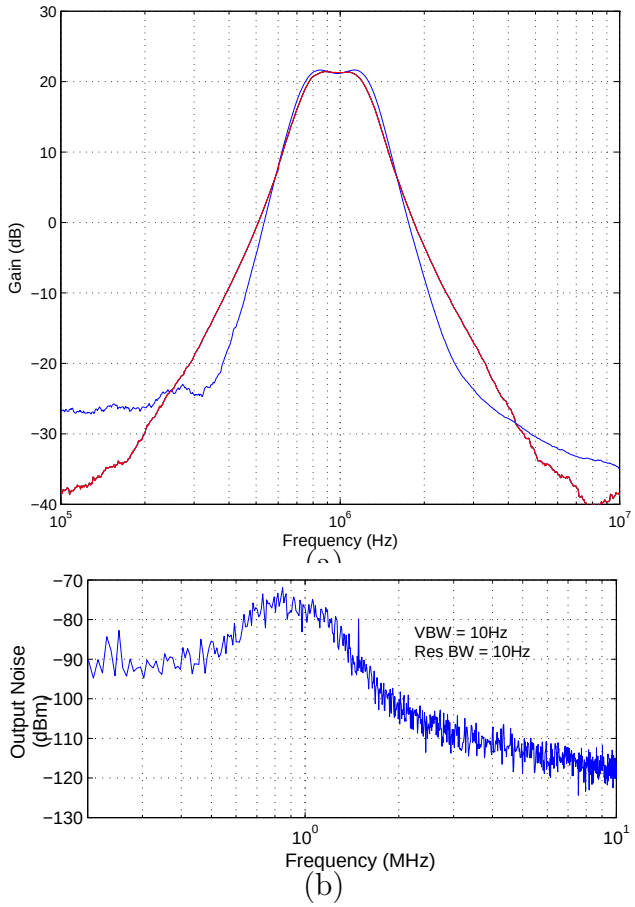


Figure 61. Magnitude response and noise spectrum of a 6^{th}- and 10^{th}-order filter.: (a) Measured magnitude frequency response of a 6^{th}- and 10^{th}-order filter designed using 2^{nd}-order sections. (b) Plot showing output referred noise spectrum for the 10^{th}-order filter.

linearity increases from -8.5dBm to -5dBm as V_{bias} is decreased from 3.3V to 1.9V. This increase in linearity comes at the cost of lowering of the low frequency corner due to the source degeneration effect. Thus, the current, I_2, needs to be programmed to a higher value than before to get the same lower time constant.

Figure 59 shows the output-referred noise measurement of the 2^{nd}- and 4^{th}-order filter sections for various programmed corners. The noise spectrum looks like the frequency response of the tuned filter, as expected from (84). Figure 59 also shows that overall noise spectrum decreases as the programmed center frequency is increased. This can be attributed to the 1/f component of the noise spectrum. The measured

Table 2. Performance Summary for the C^4 filter

Parameter	2^{nd}**-order**	4^{th}**-order**	10^{th}**-order**
Frequency Range	100Hz-10MHz	100Hz-10MHz	N/A
Q range	< 9	< 72	N/A
Output Noise (dBm @ 1MHz) (VBW = 1Hz)	-100dBm	-84dBm	-78dBm (VBW = 10Hz)
Total Power (with buffers)	0.1nW-15μW	0.25nW-15μW	20μW @ 1MHz
SNR @ 1MHz	86dB	72dB	55dB
Area	2.1e3μm^2	4.8e3μm^2	13.2e3μm^2
Programming % error	$< \pm 0.2\%$	$< \pm 0.2\%$	$< \pm 0.2\%$
Programming mechanism	Hot–electron injection	and	Fowler-Nordheim tunneling
Tunneling Voltage (V)	15	15	15

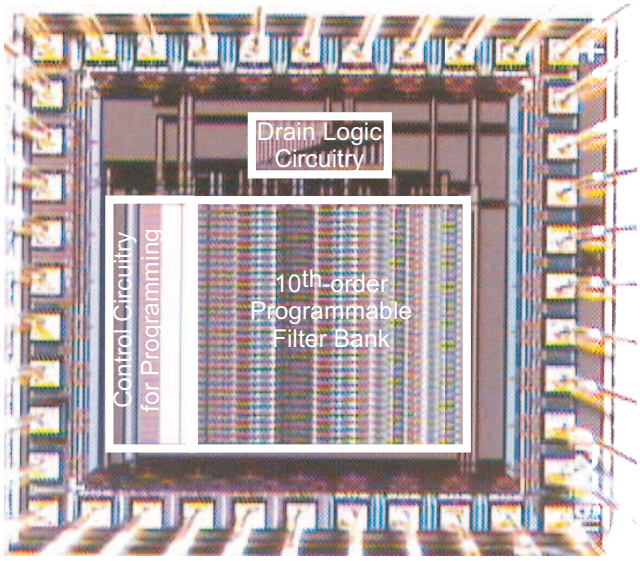


Figure 62. Micrograph of the 10^{th}-order filter-bank chip.: Chip micrograph of filter-bank chip, with 16 filters, that was used to measure the 6^{th}- and 10^{th}-order filter response. The chip includes logic and control circuitry that is used for programming. The area of the entire chip was 1.1 mm^2.

output spot-noise at 1MHz for the 2^{nd}-order section was found to be -100 dBm (using VBW = 1Hz).

6.4 High-Order Filter Implementation

We used the 2^{nd}-order section, discussed above, in cascade to implement higher-order filters. Figure 48 shows the block diagram of the 10^{th}-order filter using these core 2^{nd}-order sections. These higher-order filters can also be tuned to desired transfer functions, such as Butterworth and Chebyshev, after the circuit has been fabricated. The 2^{nd}-order sections were designed such that the Q_{max} (76) is greater than that required by the higher-order filter specification. The coefficients can be set by accurately programming the floating-gate currents. As evident from the schematic in Fig. 48, the input capacitance changes with frequency. Figure 60 shows the dependance of input capacitance with frequency for different values of C_W. This becomes a problem when these sections are cascaded. As value of C_W is increased such that it becomes the dominant capacitance, this frequency dependance goes down. But this comes at the cost of area and speed performance, and is impractical for filter-bank applications due to area constraints. A unity-gain buffer was introduced between each stage, as shown in Fig. 48, to take care of varying input capacitance without increasing C_W. The buffer was designed to have a good frequency response and linearity and thus, had no effect on the performance of the system.

Figure 61 shows the frequency response of a 6^{th}- and a 10^{th}-order filter tuned to have a center frequency of 1MHz. These filters can be tuned to have different center frequencies. The limitations in the measurement for high frequency was once again the output driving buffer. The designed 10^{th}-order filter was compact and power efficient. This filter can be used in a variety of filter-bank applications [51, 45]. Figure 62 shows the die photograph of the chip with 16 filters that was used to take the measurements. This chip can be configured as a bank of 6^{th}-order or 10^{th}-order

filters depending on the application.

6.5 Conclusion

We presented a compact continuous-time ($G_m - C$) bandpass filter circuit that can be programmed to operate from 100Hz to 10MHz center frequencies. Table I summarizes the measured performance of all the filter sections fabricated. We demonstrated the characterization results for the basic 2^{nd}-order and 4^{th}-order sections designed for high Q's. The experimental results presented were from a 0.5μm double-poly CMOS process; these results scale straightforwardly to other CMOS processes. The measurements show an SNR of 86dB and 72dB, respectively, for a 2^{nd}-order and 4^{th}-order section at a center frequency of 1MHz. We obtained Q's as high as 70 from the 4^{th}-order sections. We also presented results for a 6^{th}- and 10^{th}-order filter fabricated by cascading the 2^{nd}-order sections. These filters were programmed at a center frequency of 1MHz to have Butterworth coefficients. The measured SNR was 51dB for the 10^{th}-order filter programmed at 1MHz. The low power consumption and low area make these extremely attractive for filter-bank applications [51, 45].

CHAPTER 7

CURRENT-MODE LOGDOMAIN-FILTERS

One of the major limitation of $G_m - C$ continuous-time filters is the limited linearity that they can achieve due to their inherent voltage-mode nature and dependence on transconductance of transistor for corner frequencies. To address this, we present design of current-mode continuous-time log–domain filters. Log–domain filters have recently become an integral part of family of continuous-time filters. These filters have externally linear transfer function but internally are highly non-linear. All the log-domain filters use translinear elements to do the filtering on logarithmically compressed voltage signals. The internal exponential and logarithmic non-linearities of these translinear elements are used to design filters with the possibility of wide dynamic range. Also, these filters become important in systems with low supply voltages and hence, low voltage signal swings as most of the processing is done in current-mode.

Figure 63 shows the block diagram that illustrate the basic idea behind a log–domain filter. A log–domain filter can be conceived as a circuit composed of both linear and non–linear elements, which, when placed between a log converter and an anti-log converter, will cause the system to act as a linear filter. The most important component of a log-domain filter is the translinear element. We use a multiple-input translinear element (MITE), as proposed in [54], which uses floating-gate (FG) transistors operating in subthreshold or weak-inversion. The advantage of using MITEs is that they can be easily fabricated and characterized in a standard CMOS process.

In this chapter, we discuss the design of a fully tunable second–order bandpass filters, as shown Figure 66 that was fabricated using MITEs. We present synthesis for the second–order bandpass filter from state–space description, as explained in [55]. The second order sections can be used to then design higher order bandpass filters by cascading these second order sections. Also, these higher order filters can be

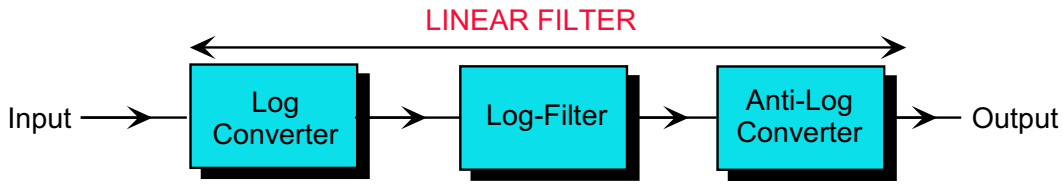


Figure 63. Block diagram of a log-domain filter.: Block diagram showing implementation of a log–domain filter. The signal is compressed into the log–domain and then the filtering is performed. The output signal is then converted back using an anti–log block.

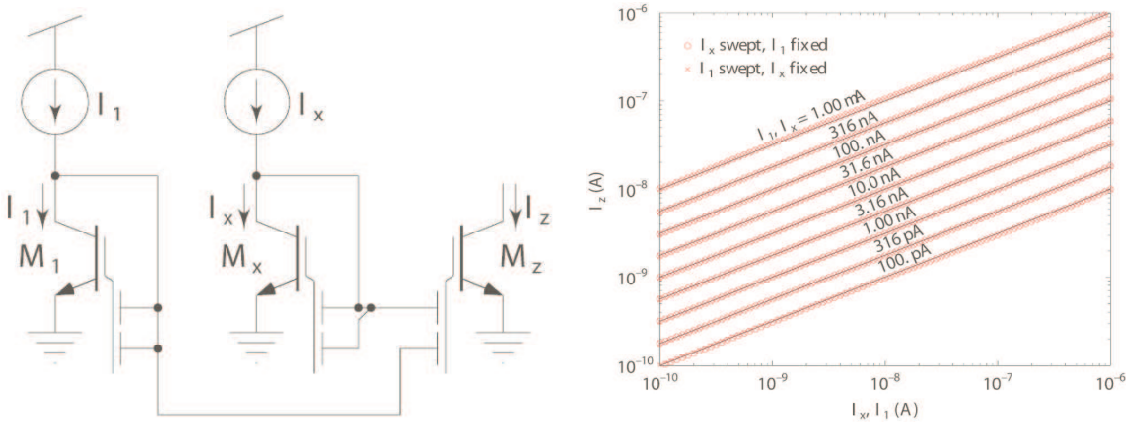


Figure 64. Square–root circuit implemented using MITEs: (a) Circuit schematic implementing the square–root function using NMOS MITEs. (b) Measured results for the implemented square–root circuit [13]. As can be seen, current–mode MITE circuits give decades of linearity in terms of signal swing.

synthesized from the state-space methods as described in [56]. The use of FGs help in making these filters tunable to get the desired frequency response and quality factor, Q, along with correcting for any mismatches after fabrication. This becomes extremely important in design of log-domain bandpass filters, which require current subtraction to get the bandpass response.

7.1 Multiple Input Translinear Elements

The multiple–input translinear element (MITE) is a device that produces an output current that is exponential in a weighted sum of its input voltages [54]. Such devices can be implemented by multiple-input floating-gate transistors operating in

sub-threshold, as shown in Fig. 65(a). The output current I is given by

$$I = I_s e^{(k_1 v_1 + k_2 v_2 + ... + k_n v_n)/U_T} \tag{99}$$

where I_s is a pre-exponential scaling current, k_n is a dimensionless positive weight, V_n is the nth input voltage, and U_T is the thermal voltage, kT/q. Individual MITEs can be networked together to construct low-power translinear circuits, called MITE networks. These networks can implement static or dynamic, linear or nonlinear systems [54]. Figure 64(a) shows circuit schematic, using NMOS-based MITES, to compute square–root [13].

$$I_z^2 = I_x I_1 \tag{100}$$

Figure 64(b) shows the measured results for the circuit shown in Fig. 64(a). As can be seen, log–domain circuits give decades of linearity in signal swing. The programmable PMOS MITE that we use to implement the second order log-domain filter is shown in Fig. 65(b). Apart from the other benefits mentioned in previous chapter, having a cascode transistor makes this structure simpler to program, since during programming we can isolate the MITE by simply setting $V_{pcascode}$ to V_{DD} and $V_{ncascode}$ to ground. Given this architecture we are then able to synthesize a function entirely in the same row or column of an array to maintain control of the gate line for programming.

The bias current sources $I_{\tau 1}$ and $I_{\tau 2}$ (see Fig. 66) are produced by a floating-gate transistor, and mirrored through a NFET cascode current mirror (see Fig. 65(c)). Through the use of a PFET floating gate transistor, we are able to accurately fix the output current to any desired level by applying the same programming techniques that are used for the MITEs.

7.2 Synthesis of Logdomain Filters

In this section, we will discuss synthesis of logdomain filters using state–space method [57]. Figure 67 shows the building block for implementing log–domain filters. The

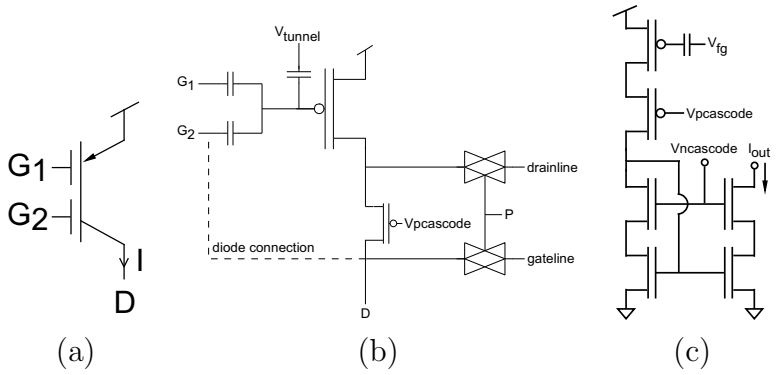


Figure 65. Schematic of Multiple-Input Translinear Element (MITE) and cascode current source: (a) MITE circuit symbol. The output current I is the exponential of a weighted sum of the input voltages. (b) Fabricated MITE structure. This structure entails the actual layout of the MITE, illustrating the PFET cascode and T-gates. This allows for the MITE to be isolated in programming mode. (c) The input bias currents are generated using this NFET cascoded current mirror in conjunction with a PFET floating-gate transistor. The programming ability of the floating-gate transistor allows for precise current levels.

two MITE elements have two inputs each of weight w. The relationship can be easily derived using 99 as

$$I_{out} = I_{in} e^{w(V_2 - V_1)/U_T} \tag{101}$$

Thus, the output current is equal to the input current scaled by a gain that is the exponential difference between the two input voltages V_2 and V_1. This is similar to other log–domain filter blocks except for a sign difference. Any MITE log–domain filter can be realized from the state–space description of the systems. A simple way to derive such a description is from desired transfer function of the filter. The logarithmic mapping between the voltage–state variable (of a voltage–mode filter transfer function) and the current–state variable that are derived from them constrains the input currents and the current–state variable to be strictly positive. After deriving the state–space description, the MITE log–domain filter can be easily implemented using the circuit structures shown in Figure 68. Figure 68 shows circuit structures for all possible terms that can occur in a set of coupled first–order linear ordinary differential equations [57]. The implemented filter structure can be simplified by eliminating unused outputs and sharing the diode–connected transistors with same inputs.

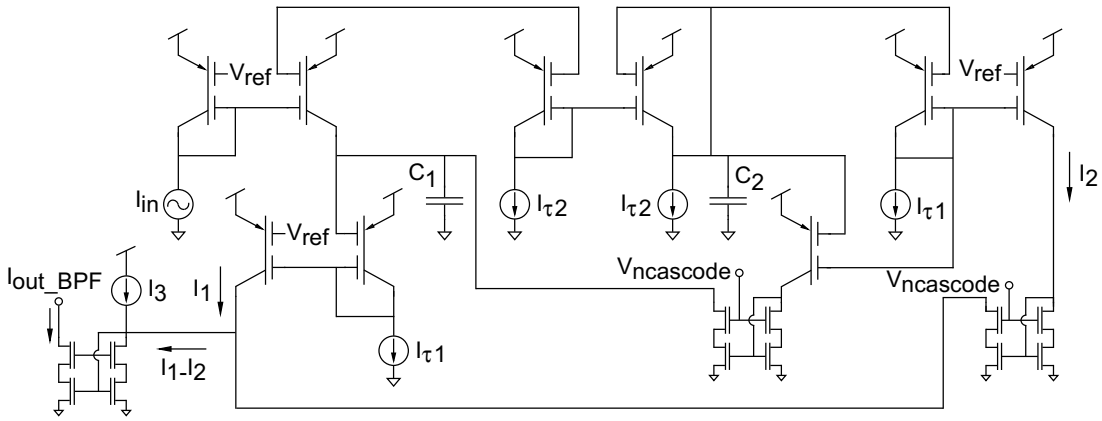


Figure 66. Schematic of second-order log-domain bandpass filter: MITE implementation of second order log-domain bandpass filter. 2-input MITEs are used. The current sources $I_{\tau 1}$ and $I_{\tau 2}$ are generated by programmable current sources.

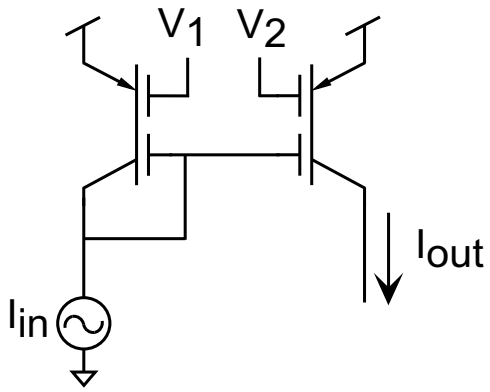


Figure 67. Log–domain filter building block.: Building block circuit for MITE log–domain filters.

The filter output can be generated by forming a weighted sum of current–state variable. The weighted multiplication can be either done by sizing the MITEs or by programming the bias currents of MITEs, as would be done in realizing the current–mirrors. We use the latter to provide the weight coefficients. One scaled, these currents can be properly added using KCL. Figure 66 shows the log-domain bandpass filter using programmable MITEs using the synthesis and simplification procedure.

The transfer function of a second order log-domain bandpass filter is given by:

$$\frac{I_{out,BPF}}{I_{in}} = \frac{sA}{s^2\tau_1\tau_2 + s\tau_1 + 1} \tag{102}$$

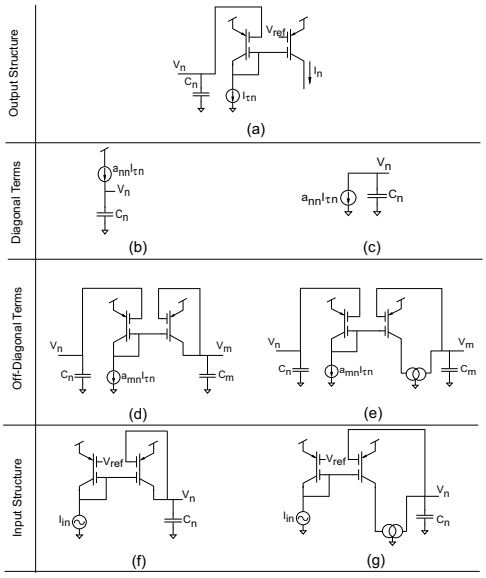


Figure 68. Circuit structures for state–space terms.: The circuits shown implement the following state–space terms [55] (a) $\tau_n \frac{dI_n}{dt} = ... \ \tau_n = \frac{C_n U_T}{\kappa I_{\tau n}}$ **(b)** $\tau_n \frac{dI_n}{dt} = ... + a_{nn} I_n - ...$ **(c)** $\tau_n \frac{dI_n}{dt} = ... - a_{nn} I_n - ...$ **(d)** $\tau_m \frac{dI_m}{dt} = ... + a_{mn} I_m - ...$ **(e)** $\tau_m \frac{dI_m}{dt} = ... - a_{mn} I_m - ...$ **(f)** $\tau_n \frac{dI_n}{dt} = ... + I_{in} - ...$ **(g)** $\tau_n \frac{dI_n}{dt} = ... - I_{in} - ...$

where A: the mid-band gain τ_1, τ_2 : time constants set by the dominant capacitances in the circuit similar to the way as in $G_m - C$ filters. Using these time constants, the center frequency and the quality factor, Q, for a bandpass filter can be defined as:

$$\tau = \frac{1}{\omega} = \sqrt{\tau_1 \tau_2} \tag{103}$$

$$Q = \sqrt{\frac{\tau_2}{\tau_1}} \tag{104}$$

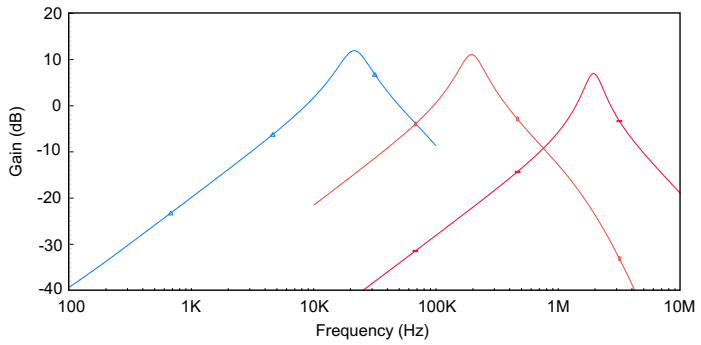
Equation () can be represented in time domain as:

$$\tau_1 \frac{dI_1}{dt} = I_{in} - I_2 \tag{105}$$

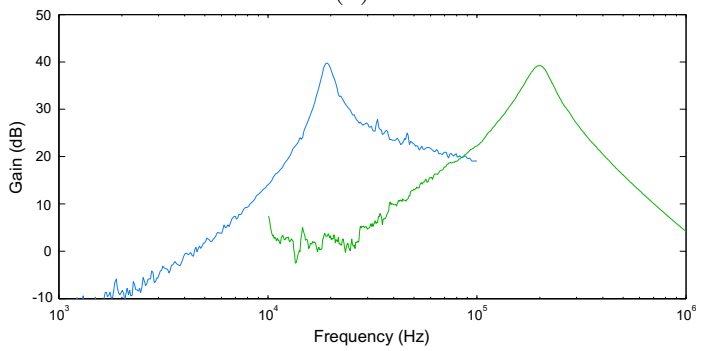
$$\tau_2 \frac{dI_2}{dt} = I_1 - I_2 \tag{106}$$

$$I_{out,BPF} = \frac{dI_2}{dt} = \frac{I_1 - I_2}{\tau_2} \tag{107}$$

where I_1 is a temporary variable. Based on the above equations and the synthesis method to implement first order systems, the second order system can be implemented as shown in Figure 66.



(a)



(b)

Figure 69. Wide range frequency tuning measurement: (a) Simulated corner frequency tuning illustrating the wide range of possible frequencies.(b) Measured corner frequencies. The corner frequencies are tuned to 20kHz and 200kHz.

7.3 Second–Order Logdomain Bandpass Filter

For the circuit schematic shown in Fig. 65 (d), the time constants are given by (using the same notation as in [57])

$$\tau_1 = \frac{C_1}{g_{\tau_1}},\ \tau_2 = \frac{C_2}{g_{\tau_2}}, \tag{108}$$

where g_{τ_1}, g_{τ_2} are transconductances of the MITEs with currents I_{τ_1} and I_{τ_2}, respectively. The bias current sources $I_{\tau 1}$ and $I_{\tau 2}$ (see Fig. 65 (d)) are produced by a floating-gate transistor, and mirrored through a NFET cascode current mirror (see Fig. 65 (c)). Through the use of a PFET floating gate transistor, we are able to accurately fix the output current to any desired level by applying the same programming techniques that are used for the MITEs.

As can be seen from 108, τ_1 and τ_2 depend on the transconductances of the

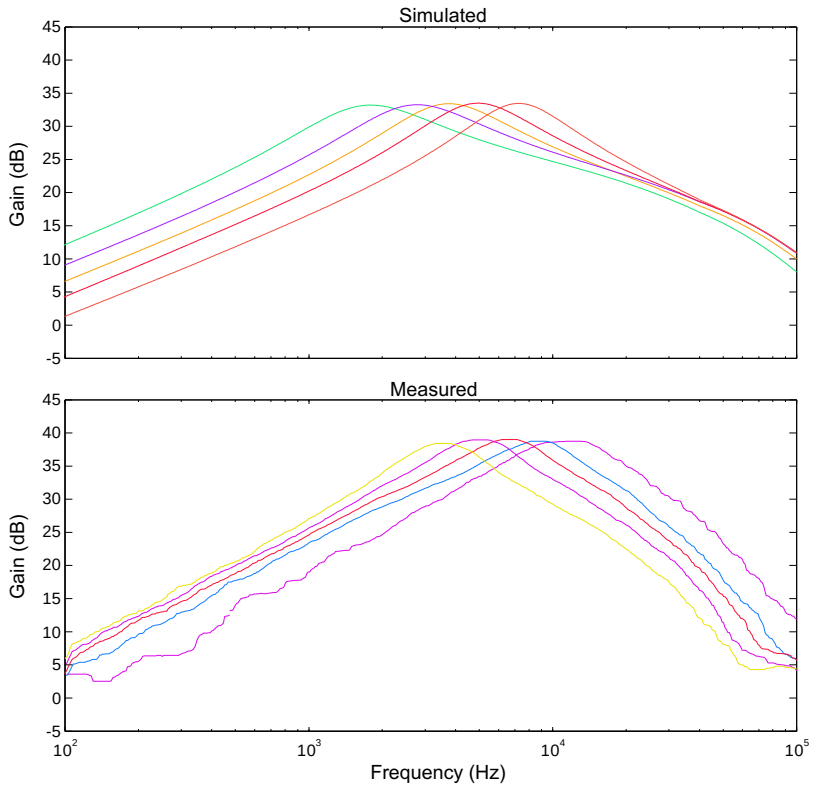


Figure 70. Measurement showing frequency tuning: The corner frequencies are modified through the variation of $I_{\tau 1}$ and $I_{\tau 2}$. Simulated and actual results are plotted for comparison.

MITEs. Thus, changing these time constants will change the center frequency and Q of the filter section. Also, the response is sensitive to the current subtraction done at the output. Thus, any mismatch in the current sources or in the MITEs currents due to fabrication gradients will have a detrimental effect on the response. The weak inversion operation of MOS further aggravates the problem of current mismatch. The ability to program these FG elements takes care of any such mismatches. This along with the already mentioned features of MITEs in [57] makes this structure a suitable candidate for log-domain bandpass implementation. Using cascodes further makes MITEs more robust to process parameters along with the additional benefits in the programming logic, as mentioned in the above section. Thus, by programming the current sinks I_{τ_1} and I_{τ_2}, the desired frequency response and Q for the bandpass filter

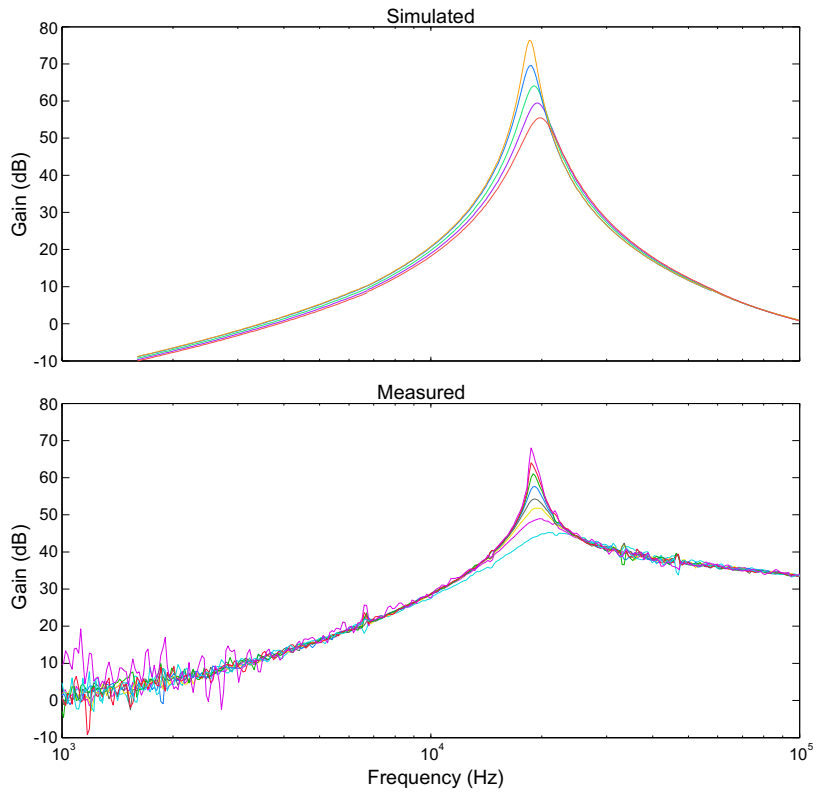


Figure 71. Measurement showing Q-tuning: Q can be increased or decreased by varying the bias currents $I_{\tau 1}$ and $I_{\tau 2}$ and bias voltage V_{ref}. Simulated and actual results are plotted for comparison.

can be obtained.

The synthesis procedure used to generate the second order bandpass filter can be generalized to obtain a circuit schematic for a higher-order bandpass filter by decomposing the n-th order system in n different first order equations. Also, higher order bandpass filter can be made by cascading the second order sections discussed above.

7.4 Experimental Results

Simulation plot showing that the designed filter can be tuned over a wide range of frequencies is shown in Figure 69. Figure 69 shows the experimental result showing the response at 20 KHz and 200 KHz, which matches with the corresponding simulations. The measurements were limited due to the setup as will be explained in the next

section.

Figure 70 shows the simulation and measured bandpass response for the programmed log-domain filter over a range of bias currents to give different corner frequencies. The frequency response agreed well with the simulation results.

Figure 71 shows the results of the experiment to get different values of Q at the same corner frequencies. In this experiment, I_{τ_1} and I_{τ_2} were programmed such that the corner frequency was kept constant while changing the Q-peak only. This was done by programming the currents to have a constant product but an increasing ratio of $\frac{I_{\tau_1}}{I_{\tau_2}}$ to increase the Q-peak. The experiment shows that a Q-peak of up to 15 can be obtained from the designed filter.

7.5 Measurements

Since log-domain filters have current as input and output, care must be taken while testing them. The measurements depend a lot on the dynamic range and frequency response of the voltage to current conversion block at the input and vice versa at the output. In the initial setup, to perform the measurements presented here, the protoboard was used that had a large capacitance and a poor noise performance. This limited the measurements as the performance of the discrete Op-amps, used to build the input and output blocks, deteriorated due to the board.

The voltage to current conversion block at the input was implemented using a discrete Op-amp in negative feedback to generate input current. The input amplitude was limited to keep the input current as linear and distortion free as possible. In the next chapter, we will present design of high–linearity input and output circuitry that can be used to measure performance of various building blocks.

7.6 Summary

We discussed a programmable second–order log-domain bandpass filter implementation using MITEs. We presented experimental results showing frequency and Q-tuning for the circuit fabricated in $0.5\mu m$ double-poly CMOS process. The experimental results agreed with the simulation plots. The second–order sections gave Q values of up to 15. These basic second order sections can be used to build programmable higher order log-domain filters using MITEs.

CHAPTER 8

MEASUREMENT ISSUES

Testing is an important part of designing analog systems operating at IF band frequencies. This becomes even more critical when the signals are current–mode. Taking clean measurements of the performance of these circuits involves the following:

- Measurement circuitry (on-chip) around the main circuit blocks to be able to drive the output signal off-chip cleanly at relatively high speeds (1-100MHz).
- Off-chip instrumentation to take clean high speed measurements for verification purposes.

This implies designing voltage–buffers with good linearity and frequency response to drive off–chip loads, linear V-to-I and I-to-V converters for current–mode signals and designing printed circuit boards (PCBs) to measure the performance. The instrumentation required to do these measurements include network analyzers and spectrum analyzers that would operate over a range from 1KHz-100MHz. Presently, we do not have a single equipment that covers the entire range. We used two network analyzers with ranges of 1mHz-100KHz and 30KHz to 6GHz, respectively.

8.1 Analog Voltage Buffer

Figure 72(a) shows the circuit of a buffer designed to drive high off-chip loads. The DC bias is set by the unity–gain buffer made by five transistor differential pair with current–mirror load. The unity–gain buffer is biased to have a real low cut-off such that it does not affect the signal that is being buffered. As evident, the designed voltage buffer does not pass DC and very low frequency components depending on the cut–off of the unity–gain setting up the bias. The gain of the buffer is decided by the capacitor ratio and can be accurately set. The capacitors should be sized to be larger

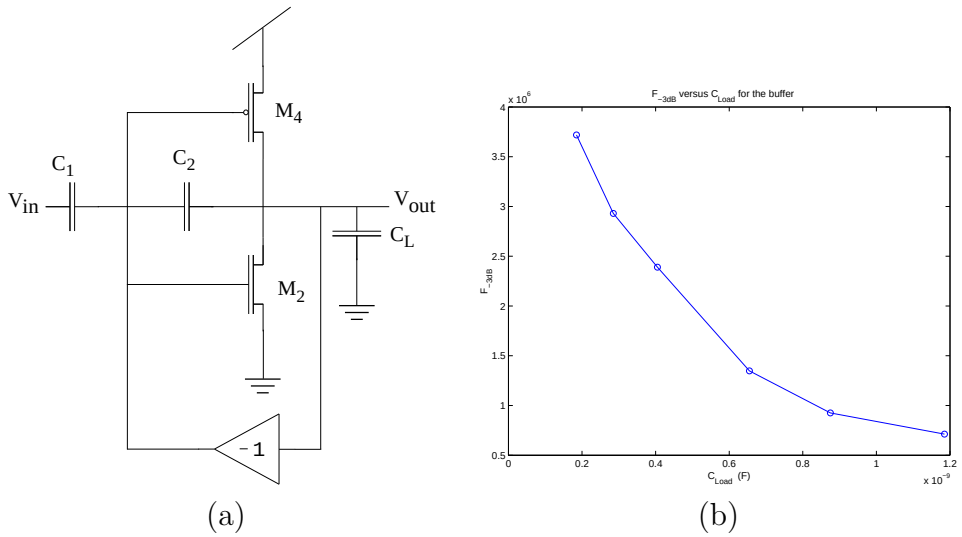


Figure 72. Analog buffer to drive off-chip loads: (a) Circuit schematic of the designed output buffer. (b) Cut-off frequencies for various capacitive loads. The frequencies were inversely proportional to the capacitance value.

than the C_{gs} of the transistors that is determined by the transistor dimensions. The (W/L) of the transistor is governed by the required cut-off frequency. The designed buffer is more linear and gives the maximum bandwidth/unit power compared to some of the existing approaches. The use of capacitors instead of resistors give better matching and also do not load the output node at low frequencies. Small–signal model was used to analyze this circuit in more details and can be easily done by the reader. Figure 72(b) shows the plot of cutoff frequencies for different loads obtained from the same measurement. The circuit has a bandwidth of about 3.7 MHz for a capacitive load of approximately 185 pF, which is very reasonable considering the fact that most printed circuit boards have a typical capacitance between 10-20pF.

8.2 Linear I-V and V-I converter

Current-to-Voltage $(I-V)$ and Voltage-to-Current $(V-I)$ converters play an important role as interface/measurement elements in current-mode mixed signal systems. Compact $I-V$ and $V-I$ converters are essential in realizing the high performance offered by current-mode systems. Specifically, it is important that these interface

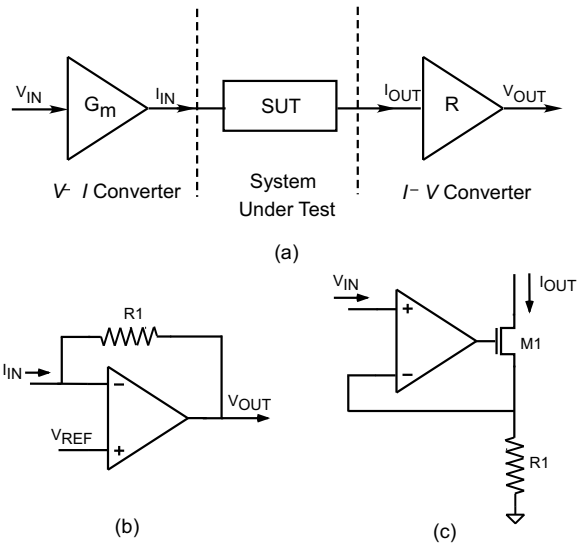


Figure 73. Interface circuitry for current-mode systems: (a) Block diagram for measurement and characterization of current-mode systems. (b) Transimpedance amplifier used for $I-V$ conversion. (c) Typical circuitry used for $V-I$ conversion.

elements offer a high linear range, bandwidth and a variable conversion gain. More importantly, their performance should remain unaffected by the loading effects of current-mode systems.

A popular approach to implementing $I-V$ converters is to configure an operational amplifier as a charge integrator. This approach, owing to sampling delays is limited to measuring low frequency currents. A transimpedance amplifier, as shown in Figure 73(b) provides continuous time $I-V$ conversion and is a viable alternative. This approach requires careful consideration to compensation to ensure good performance [58]. Also, measuring small currents on chip is prohibitive owing to the large values of resistors needed. Logarithmic converters using BJTs have a high dynamic range but implement a non-linear current conversion and are not suited for standard digital CMOS processes.

$V-I$ converters play a vital role at the input interface of current-mode systems. A common approach to current generation involves the use of an operational amplifier with a MOS transistor $M1$ and a resistor $R1$ as shown in Figure 73(c). Negative

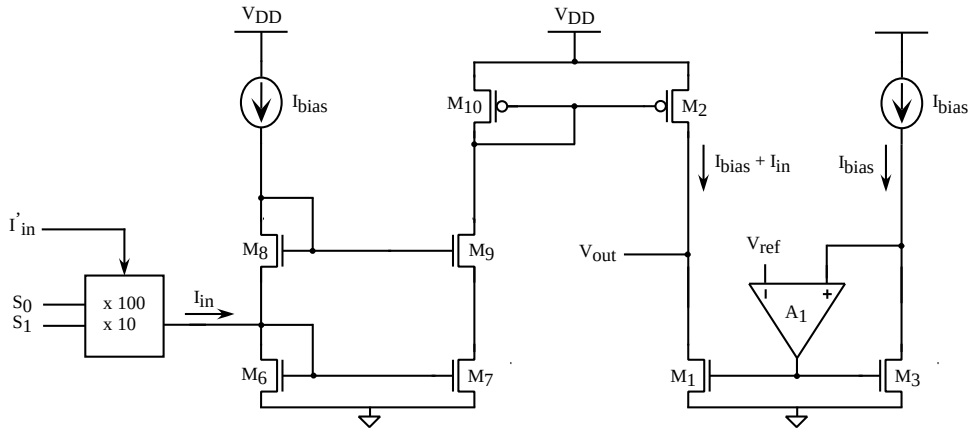


Figure 74. Circuit Schematic of the proposed $I-V$ converter: Transistors $M1-M2$ perform the core $I-V$ conversion while amplifier $A1$ serves to set the DC equilibrium for the high gain output voltage. Switches S_0 and S_1 implement input current multiplications of 100 and 10 respectively to increase the linear range.

feedback ensures that the current through the transistor $M1$ is equal to the applied input voltage divided by the resistor $R1$. For a given size of $M1$ and resistor $R1$, the finite rail-to-rail output voltage swing of the amplifier poses the major limitation to the achievable linear range of currents. Alternate approaches that have been proposed for $V-I$ converters [59], [60], [61], [62] suffer from limited linearity and/or susceptible to loading conditions affecting performance.

We propose easy to design, high performance, compact interface circuits to aid in the interface/measurement of current-mode systems. The proposed $I-V$ converter, uses the output impedance of MOS transistors to perform the current conversion [63]. The key issue in using such an approach is the difficulty of biasing the high-gain output node. This is addressed through the use of negative feedback and replica biasing. The $V-I$ converter described in this paper is compact, easy to design and uses a single external resistor to set its transconductance. The design adopted in this paper is an improvement over that in [62]. This makes the performance of the $V-I$ converter immune to loading conditions and experimental results are presented as well.

8.2.1 Current-to-Voltage Converter

Figure 74 shows the circuit schematic of the proposed $I-V$ converter that consists of the core converter, the replica biasing scheme and the current multiplication block that provides current ranging capability. The $I-V$ conversion is performed using transistors $M1$ and $M2$ where transistor $M2$ is a common source amplifier with $M1$ being the active load. For no signal input, the DC operating point for the high gain output voltage, V_{out} is designed to equal V_{ref} through the use of replica transistor $M3$, identical current source I_{bias} and the operational amplifier $A1$. On account of negative feedback, the amplifier $A1$ sets the gate of M3 such that at a drain voltage of V_{ref}, its drain current equals I_{bias}. This ensures that the drain voltage of $M1$ equals V_{ref} as well.

An input current I_{in}, is mirrored through current mirrors $M6/M7$ and $M10/M2$ such that a drain current of $I_{bias}+I_{in}$ flows through $M2$. Since, the current through $M1$ is set to equal I_{bias}, the difference current ΔI_{in} causes a change in the output voltage,(ΔV_{out}) given by,

$$\triangle V_{out} = (r_{o1} \| r_{o2}) \triangle I_{in} = r_o \triangle I_{in} \tag{109}$$

where r_{o1} and r_{o2} are the output impedances of transistors $M1$ and $M2$ respectively. It should be noted that the conversion gain is set by the output impedances of transistors $M1$ and $M2$ and can be designed to be quite large. Also, to a first approximation, the $I-V$ conversion given by (1) is linear.

The non-linearities and hence the distortion introduced can be estimated by utilizing the relationship between the drain current of a transistor and its output impedance. Assuming, a first order MOS model, the change in the output voltage, is given by,

$$\triangle V_{out} = \left[\frac{1}{\lambda_1 I_{bias}} \| \frac{1}{\lambda_2 (I_{bias} + \triangle I_{in})} \right] \triangle I_{in} \tag{110}$$

Assuming, that the λ's of $M1$ and $M2$ are equal and further assuming that the signal

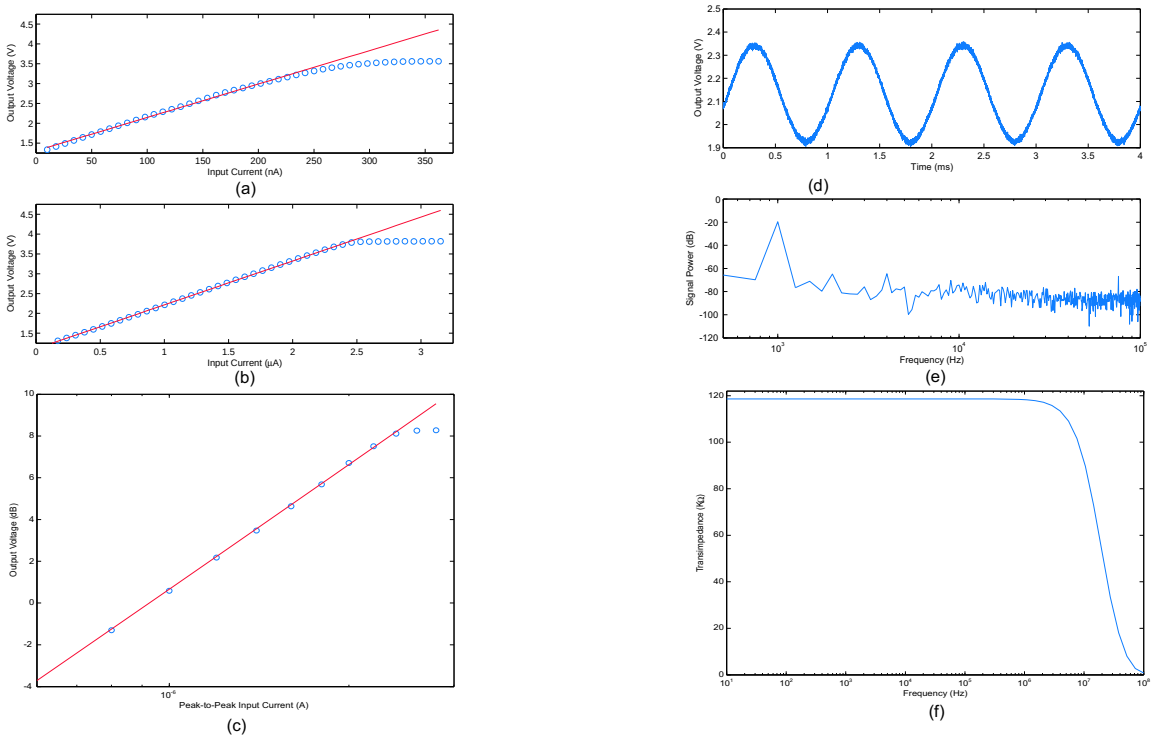


Figure 75. Measured results for the proposed $I-V$ converter: *The solid line represents the theoretical linear fit and the o's represent the measured data points.* (a) DC sweep with the internal current gain set to 100 with switch S_0 set low and S_1 set high. (b) DC sweep with the internal current gain set to 10 with switch S_0 set high and S_1 set low. (c) Measured plot that shows the $1-dB$ compression point that occurs at an input signal amplitude of $1.3\mu A$. (d) Output voltage of the $I-V$ converter for an input sinusoidal current with an amplitude of $0.2\mu A$ at a frequency of $1KHz$. (e) FFT of the output voltage showing a second harmonic that is $40dB$ lower than the fundamental and a THD of 0.82% . (f) Post-Layout simulation of the frequency response of the $I-V$ converter showing a bandwidth of $10MHz$ and a tranimpedance gain of $118K\Omega$ that agrees closely with the measured value of $105K\Omega$.

current is much smaller than the bias current I_{bias}, (2) simplifies to,

$$\triangle V_{out} = \frac{\triangle I_{in}}{2\lambda I_{bias}}\left[1 - \frac{\triangle I_{in}}{2I_{bias}}\right] = \triangle I_{in} r_o - \frac{\triangle {I_{in}}^2 r_o}{2I_{bias}} \tag{111}$$

From (3) it is clear that the second harmonic term and hence the distortion is proportional to the input signal amplitude and is inversely proportional to the bias current. This brings about a direct tradeoff between distortion and power dissipation. A differential approach can help eliminate the even order harmonics and lead to lower levels of distortion.

The $I-V$ converter can be approximated to be a single pole system with the

dominant pole being at the output node. The small signal bandwidth is given by,

$$f_{-3dB} = \frac{1}{2\pi r_o C_o} \tag{112}$$

where, C_o is the total capacitance at the output node. It must be noted that the bandwidth of the $I-V$ converter is inversely proportional to the gain. Therefore, for a given gain, minimizing the parasitic capacitance at the output node maximizes the bandwidth. For the same reason, the output of the $I-V$ converter must be followed by a voltage buffer.

The current multiplier block serves to increase the dynamic range of the $I-V$ converter. Switches $S_0 - S_1$, when turned on, multiply the input current by a factor of 100 and 10 respectively. The current multiplication block consists of a single $1:10$ current mirror and a pair of $1:10$ current mirrors cascaded together. The current mirror is designed using cascodes such that accurate mirroring is achieved. Also, the linear range can be further increased by providing higher multiplication ratios.

Figures 75(a) and 75(b) show the measured DC transfer curves for the $I-V$ converter for two different current multiplication switch positions of 100 and 10. The $I-V$ converter exhibits a current range of $10nA$-$2.5\mu A$ that translates to a linear range of 2.4 decades. The current gain and hence the output impedance of the $I-V$ converter can be measured from the slope of the DC transfer curves and has been estimated to be $105K\Omega$. Using this measured value of r_o and the extracted layout parasitic capacitance of $C_o = 140fF$, the small signal bandwidth of the $I-V$ converter can be estimated to be about $10.8MHz$. Figure 75(f) shows the simulated small signal frequency response that is in close agreement with the estimated bandwidth.

Figure 75(c) shows a plot of the peak-peak output voltage of the $I-V$ converter for different input current amplitudes. The $1-dB$ compression point of the $I-V$ converter is measured to be at an input current amplitude of $1.3\mu A$. The measured transient response of the $I-V$ converter for a sinusoidal current input at an amplitude of $0.2\mu A$ and a frequency of $1KHz$ is shown in Figure 75(d). The FFT of the output

is shown in Figure 75(e). The measured THD is 0.82% while the worst-case THD is 3.9% at the $1-dB$ compression point.

8.2.2 Voltage-to-Current Converter

Figure 76(a) shows the circuit schematic of the CMOS $V-I$ converter. The use of amplifier $A2$ helps fix the output node at a fixed voltage thereby nullifying the effect of the output capacitance leading to a high bandwidth. This also serves to isolate the output of the $V-I$ converter from external loading effects.

Figure 76(b) shows the detailed schematic of the feedback amplifier $A1$. The use of a regulated cascode current mirror ($M1-M4$) ensures that the drain of $M1$ is set to a well defined value of V_{ref}. Also, the regulated cascode increases the output impedance of the current mirror and the matching between the drain currents of $M1$ and $M2$. With the drain of $M1$ set to V_{ref}, the output current I_{out} of the $V-I$ is,

$$I_{in} = \frac{(V_{in} - V_{ref})}{R_{in}} = I_{out} \tag{113}$$

where, V_{in} is the applied input voltage and R_{in} is the value of the resistor used.

The small signal input impedance ($r_{o,CL}$) at the drain of $M1$ is given by,

$$r_{o,CL} = \frac{1}{g_{m3}(1 + A + g_{m1}r_{o1}g_{m3}r_{o3})} \tag{114}$$

where A is the open loop gain of the feedback operational amplifier $A1$, and g_{m3} is the transconductance of the cascode transistor $M3$. The use of a regulated cascode, ensures a very low impedance at the drain of $M1$ that further ensures that the voltage at the drain remains at V_{ref} independent of the current flowing through $M1$. This ensures that (5) holds for a large range of currents.

With proper design and a correct choice of resistor R_{in}, the linear range of the $V-I$ converter will usually not be an issue. There are however two key factors that need to be considered. Assuming the feedback amplifier $A1$ to be ideal, the gate-source voltage of $M4$ can reach a value of $V_{dd} - V_{ref}$ at most and thereby places an upper

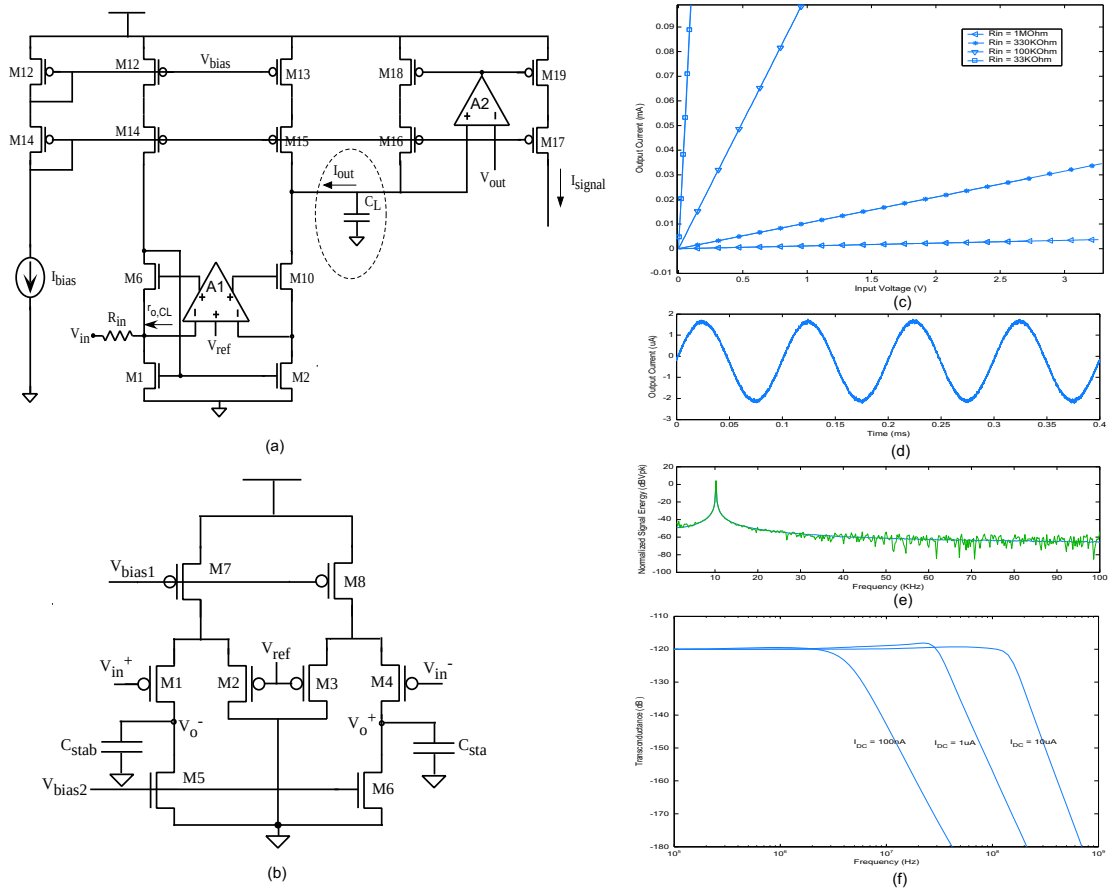


Figure 76. Circuit schematic and measured results for the $V-I$ converter: (a) Schematic of the rail-to-rail CMOS $V-I$ converter. (b) Circuit schematic showing the operational amplifier that is used in the regulated cascode loop of the $V-I$ converter. A capacitance of $100fF$ is added to the output of the operational amplifier to ensure stability. (c) DC sweep of the $V-I$ converter for different values of input resistance R_{in}. V_{ref} was subtracted to clearly illustrate the effect of R_{in} on the slope of the conversion. The $V-I$ converter displays an input voltage swing greater than the power supplies. (d) Output transient response of the $V-I$ converter for an input voltage signal of $3.3V_{pp}$ at $10KHz$. (e) FFT of the input and the output signal waveform showing clearly that the $V-I$ converter does not introduce additional non-linearities and is highly linear. (f) Simulation showing the frequency sweep of the $V-I$ converter with a DC bias current of $1\mu A$ and a bandwidth of $30MHz$. As expected, the cut-off frequency is high because it is limited by only the parasitic capacitances. Also shown are simulation results for bias currents of $100nA$ and $10\mu A$.

bound on the output current. Also, the pFET current mirrors come out of saturation and lead to distortion when the gate-source voltage of $M1$ reaches $V_{dd} - 2V_{dsat,p}$. This leads to an an upper bound on the linear range as well.

The input voltage swing for the $V-I$ converter is not limited by the power supply and can therefore exceed the positive supply voltage. When the input signal V_{in}, falls below V_{ref}, signal inversion occurs. The output current in this case is limited by the bias current of the PMOS transistors and should therefore be designed accordingly. The speed of the $V-I$ converter is dependent upon the DC bias current, I_{bias} and the parasitic capacitances at the output. The regulated cascode loop must be designed such that the loop bandwidth is greater than the input signal bandwidth and its stability must be ensured as well.

Figure 76(c) shows the measured DC transfer curves for the $V-I$ converter for different values of R_{in}'s. As expected, the slope of the curve is dependent upon the value of R_{in} and the transfer function is truly linear. The input voltage can swing greater than rail-to-rail and the output currents display over 5 decades of linear range. Figure 76(d) shows the measured transient waveform for the $V-I$ converter with an input voltage swing of $3.3V_{pp}$ and an input resistance R_{in} of $1M\Omega$ at $10KHz$. As can be seen from Figure 76(e), the FFT of the output of the $V-I$ converter looks identical to the input thereby introducing no distortion. Figure 76(f) shows the AC performance of the $V-I$ converter. For a DC bias current of $1\mu A$ the $V-I$ converter displays a $30MHz$ bandwidth.

8.3 Summary

In this chapter, we presented the design of interface circuitry that is used to measure the performance of analog systems. We presented a high linearity, high speed analog voltage buffer. We also proposed linear $I-V$ and $V-I$ converters that are compact and easy to implement in a standard digital CMOS process. The proposed circuits

have been implemented in a $0.5\mu m$ CMOS technology and experimental results have been presented. Both the $I-V$ and $V-I$ converters display a large linear range and introduce very low distortion. The $I-V$ converter has a bandwidth of $10MHz$, 2.4 decades of linear range and a THD of 0.82%. The $V-I$ converter has 5 decades of linear range, a $30MHz$ bandwidth and introduces virtually no additional non-linearities.

CHAPTER 9
APPLICATIONS, IMPACT AND FUTURE RESEARCH

9.1 Impact of presented research

About two decades ago, a lot of signal–processing systems started using DSPs for the flexibility and accuracy they provided. Analog system due to their inherent sensitivity to electrical noise and lack of programmability with easy became mostly peripheral or interface components. Today with large system and application integration for portable systems, power budget is limited by the battery life. Thus, the traditional approach of performing all the processing using DSPs can be power intensive. In this thesis, we present a method for coexistence for analog and digital in order to maximize the computation ability while prolonging the battery life. The basic idea is to perform as much processing as possible in analog before using DSPs to do the rest. The advantage of this approach comes from the fact that the signals when sent across any medium are analog by nature. In this chapter, we briefly summarize the work that has been done so far with regards to the proposed CADSP approach. We presented programmable analog signal processing systems using floating–gate devices that can be realized in a power efficient way from audio band to low IF band. We presented a fully programmable analog vector–matrix multiplier architecture. This is the first current–mode architecture and is used for a practical DCT application. We also presented the first analog architecture for a programmable modulator/demodulator (PAMD) system. The impact of the presented research can be summarized as follows:

Power efficient design methods and performance of floating–gate devices: I briefly overview in Chapter 2 the basic floating–gate device along with the circuit schematic and layout. I derived expressions governing the performance of floating–gate device and compared them with the standard MOS device when driving similar loads. The derived expressions for maximum operating frequency and SNR revealed that the

performance of a floating–gate device is very similar to that of a standard MOS device. The intrinsic transition frequency, f_T, for the FG device is a factor κ less than that of a MOS device. This is due to the fact that f_t is computed by short–circuiting the output and does not take into account the loading at the output. Thus, f_T may not necessarily give the true information of the performance of a device when used in a real circuit.

I discussed the design considerations in doing a power efficient design and basic motivation behind a power–efficient design than a low power design. To get most power efficiency, the circuit should be operated close to the maximum cut-off frequency for a given current level and that can be obtained by operating circuits in sub–threshold. However, it is not always possible to operate circuits in sub–threshold. This is because for higher frequency response, the bias current level has to be increased. This implies that the device size should be increased to keep the device in sub–threshold, which increases the area and the intrinsic parasitic capacitances. The increase in parasitic capacitance hurts the frequency response and does not give the returns for burning more power. In such a case, it is desirable to design and operate circuits in moderate–inversion in order to save area and get higher performance in terms of speed.

I designed and fabricated a printed-circuit board (PCB) to interface with FPGA to perform fast and accurate programming. The board was tested and is being currently used by ICELAB to perform programming. This is a joint project with several members of ICELAB involved [20]. I briefly discussed the predictive algorithm, which was developed by Mr. Abhishek Bandyopadhyay and Mr. Gullermo Serrano, to perform fast and accurate measurement along the the PCB mentioned above. I explained the complete algorithm along with the measured results showing calibration and programming. I also discussed various applications where floating–gate devices can definitely make a big impact. One of such applications is to remove offsets due to any mismatch

permanently after the chip has been fabricated as discussed in chapter 2.

Design, simulation and testing of vector–matrix multiplier (VMM): A voltage–mode vector–matrix multiplier operating in sub–threshold was designed in our group [6]. Initial results showing proof of concept were presented. The architecture although operating in sub–threshold had poor linearity and frequency response. A voltage–mode implementation operating in sub–threshold is limited in linearity due to the exponential I-V relationship of the transistor operating in saturation. This limitation in linearity can be alleviated to a certain extent by using methods like source degeneration, which degrades the frequency response leading to a higher power consumption.

I designed, simulated and fabricated a novel 128x32 current–mode analog vector–matrix multiplier (VMM) using floating–gate devices. The presented VMM is the first programmable analog current–mode architecture and is fully–differential. In order to obtain high power efficiency, a sub–threshold implementation is ideal. I, along with Mr. Abhishek Bandyopadhyay and Mr. Venkatesh Srinivasan, analyzed the core multiplier cell for the governing design equations. The programmable VMM system can be used to implement various signal processing functions along with different kinds of block transforms. I designed a PCB to test the functionality and frequency response of the vector–matrix multiplier for different current levels [64]. I measured the frequency response of the VMM system for various bias levels from deep sub–threshold to moderate inversion validating the theory of power–efficient design presented in chapter 2.

The current–mode VMM architecture is suitable for low power applications and has a power per bandwidth ratio of 531nW/MHz per differential multiplier. For a bandwidth of less than 10MHz, this architecture is capable of performing 1 million MAC/0.9μW as compared to a commercially available DSP (TMS32005x series) that gives 1 million MAC/0.25mW. The VMM chip gives a linearity of over 2 decades with a worst case error of $\pm$2.5%. The IC prototype was fabricated in a 0.5μm

CMOS MOSIS process and occupied an area of 0.83mm^2.

We demonstrated block matrix transforms using this architecture. I, with my colleague Abhishek Bandyopadhyay, also tested the VMM to perform Discrete Cosine Transform (DCT) on an image to show the application to signal processing systems. The VMM chip can be used for applications like audio and video processing.

Design, simulation and testing of a programmable analog modulator/demodulator (PAMD) system: I designed and simulated a novel fully programmable analog modulator/demodulator (PAMD) chip. This system can be used to implement various communication schemes such as an OFDM modulator. The complete system architecture is discussed in Chapter 5. One of the most important components of the PAMD chip is an arbitrary waveform generator (AWG). AWG is based on a direct digital synthesis (DDS) architecture. The presented AWG can be used to generate any analog waveform and is based on floating–gate devices. The generated waveforms is then used to modulate/demodulate any signal. As discussed in the chapter, the phase noise of the generated waveform depends heavily on clock jitter and any random error in the programmed floating–gate value.

I tested the modulator chip with an array of 64x8 floating–gate elements along with the peripheral circuitry. Each column has 64 floating–gate elements and the chip generates four fully-differential current waveforms. The core structure was simulated to operate at a clock frequency of 50MHz. I presented the measured results showing modulation and demodulation using PAMD system in chapter 5. The output spectral purity for the waveform generator was dependent on the clock jitter and error in the programmed charge. The IC prototype was fabricated in a 0.5μm CMOS MOSIS process and occupied an area of approximately 1mm^2. The measurement is especially critical for the PAMD system as the output signal is current. The output I-to-V converter was designed carefully to be able to measure the output signal without adding any distortion.

As can be seen from the modulator die photograph, this chip can be easily extended to generate as many fully differential waveforms as needed. Increasing the column size, N, presents an interesting trade–off. On one hand increasing the column size reduces the quantization error, but it increases the chip area and requires a high speed clock to generate the same output frequency, f_{out} or f_{clk}/N. The presented PAMD architecture can be easily designed to operate at higher frequencies and does not necessarily need to operate in sub–threshold. Low–pass filtering required at the output can be performed by the line capacitance if an steep roll-off is not required. The filter characteristics depend a lot on the number of samples in each cycle (or the oversampling ratio) and has to be taken into account when designing for column size and maximum f_{out} desired.

Investigating use of floating–gate devices to design programmable OTAs: I, along with Dan Allen, simulated, designed and fabricated two implementations of programmable fully–differential Operational Transconductance Amplifiers (OTA) using novel common–mode feedback circuits. We used floating–gate devices in both OTA structures. The chip was tested with the help of Angelo Periera and Guillermo Serrano [50]. The primary advantage of FG-OTA1 as compared to FG-OTA2 was that it did not required any external circuit for common–mode feedback and thus, making it compact. It uses the same floating–gate capacitors, which were used to match the output current sources, in feedback to obtain the CMFB. These capacitors do not affect the DC gain by loading the output node at DC, yet they perform the CMFB operation all the way down to DC. The matching between N_6 and N_7, shown in Fig. 38(a), can become a real issue for this implementation as any mismatch may cause the two common–mode output voltages to move differently and may even saturate one side while keeping the other balanced. This limits the use of this configuration as such in filter implementations. This effect becomes more prominent if the output stage is cascoded to increase the DC gain.

Although FG-OTA2 has the disadvantage of consuming more area and requiring a dedicated reference to set the common–mode, it has several advantages that make it relatively easy to use in filter designs. FG-OTA2 can be easily cascoded to give a high output resistance, which decreases the dominant pole of the OTA-C block, giving it a more ideal integrator behavior over a wider frequency range. The high output resistance also results in higher gain for FG-OTA2. The cascoded NMOS current mirrors reduce the channel length modulation effect when mirroring currents. The output common–mode in FG-OTA2 is externally set by V_{ref} and can be fixed to any desired voltage and does not depend a whole lot on the device properties unlike FG-OTA1. This helps in the cascading these OTAs to design higher–order filters without worrying about the common–mode of the next stage.

Due to the advantages of FG-OTA2 as compared to FG-OTA1 in terms of ease of design and performance, FG-OTA2 was used to design second–order programmable filter sections. The concept of offset–removal in single–ended OTAs was published in [49].

Investigating use of floating–gate devices to design programmable $G_m - C$ filters: I used the programmable FG-OTA2 to design and simulate fully–differential $G_m - C$ lowpass and bandpass filters. I tested the chip and the results are presented in Chapter 5. I also designed a PCB to test all these chips along with the programmable FG-OTA chips for their functionality and performance. As presented, these filters can be programmed to operate anywhere from audio band to lower MHz band after fabrication. I presented experimental results from two programmable $G_m - C$ biquads: the lowpass second–order section and the band pass biquad. Any higher order filter can be realized as a cascade of biquad filters. Although there are several ways to realize higher order filters, cascade filters are the easiest to design as well as to tune. Based on the presented measurements, it is possible to design a fully programmable higher order bandpass filter that can be tuned to different responses (like Butterworth,

Chebyshev) at different frequencies by programming appropriate coefficients.

Investigating use of floating–gate devices to design programmable $G_m - C$ filters using C^4s: Traditional $G_m - C$ filter implementations based on Operational Transconductance Amplifiers (OTAs) that were discussed in previous chapter are area-intensive, thus making them unsuitable for filter-bank applications. I designed, simulated and fabricated a new and compact bandpass $G_m - C$ second–order section using capacitively–coupled current conveyor (C^4) to operate at IF band (from 100Hz to 20MHz). This filter section is compact and power efficient. I designed a PCB to test these filters for performance. I designed the test board such that it could be used for the second–order sections as well as higher–order filter sections. I, along with my colleague Paul Smith, tested these chips. One of the chips that was tested to obtain the measurement results was designed and laid out by David Graham.

I demonstrated the characterization results for the basic 2^{nd}-order and 4^{th}-order sections designed for high Q's. The experimental results presented were from a 0.5μm double-poly CMOS process; these results scale straightforwardly to other CMOS processes. The measurements show an SNR of 86dB and 72dB, respectively, for a 2^{nd}-order and 4^{th}-order section at a center frequency of 1MHz. We obtained Q's as high as 70 from the 4^{th}-order sections. We also presented results for a 6^{th}- and 10^{th}-order filter fabricated by cascading the 2^{nd}-order sections. These filters were programmed at a center frequency of 1MHz to have Butterworth coefficients. The measured SNR was 51dB for the 10^{th}-order filter programmed at 1MHz. The low power consumption and low area make these extremely attractive for filter-bank applications [51, 45].

Investigating use of floating–gate devices to design programmable log–domain filters: I simulated and designed programmable log–domain bandpass filters using floating–gate circuits. I, with help of my colleague Mike Lo in layout, sent a chip with 2^{nd}-order bandpass filters [65]. These current–mode log–domain filters are extremely power–efficient, highly linear and compact. They can be used for frequency

ranges up till low MHz ranges. I presented experimental results showing frequency and Q-tuning for the circuit fabricated in $0.5\mu m$ double-poly CMOS process. The experimental results agreed with the simulation plots. The second–order sections gave Q values of up to 15. These basic second order sections can be used to build programmable higher order log-domain filters using MITEs. I also designed and fabricated a 6^{th}-order programmable log–domain filter by cascading the 2^{nd}-order sections.

The work presented here has been published in various conferences and will be submitted for publication in journals.

9.2 Applications

The presented systems such as VMM, PAMD and continuous–time filters can be used for various applications requiring signal–processing. All these systems process signals in analog domain and are extremely power–efficient compared to their digital counterparts. Various other applications may require slight modification to the presented architecture that can be done easily depending on the requirements.

9.2.1 Universal Block Transforms

VMM system presented in Chapter 3 can be used to perform any arbitrary 2-D transforms as it is fully programmable. The presented current–mode VMM system can also be used to perform any convolution or correlation operation for various signal processing applications. VMM system was used in the MATIA chip to perform DCT [66]. If the input waveforms are continuous, then the result is also a continuous waveform. This provides additional computational options at the output such different output signal sampling. The presented current–mode multiplier can also be used to implement basic FIR filters. The components that perform multiplication and summation in the FIR filter can be replaced by an architecture similar to that of VMM.

9.2.2 Rapid Prototyping

The presented PAMD system can be used for rapid prototyping tool for analog systems, much like a Field Programmable Gate Array (FPGA) is for digital systems. The modulator/demodulator system can also be used for some sort of communications based Field Programmable Analog Array (FPAA) that is specifically designed for processing analog communication signals. An effort in this direction is being pursued by members of ICELAB [3].

9.2.3 Chirped modulator using PAMD

As discussed, PAMD architecture can be used as OFDM modulator and demodulator. We discussed possible implementation architectures for OFDM in Chapter 4. The columns in the AWG can also be programmed to generate arbitrary waveforms, which can find its application in many other areas such as generating chirped waveforms to perform pulse modulation. The possibility of generating any waveform with similar design enables a variety of communication schemes that are expensive to implement for portable applications.

9.2.4 Cochlear implant and audio processing

One of the primary applications of the C^4 lies within a bank of filters. By placing these bandpass filters in parallel, the C^4s perform a frequency decomposition of any incoming signal. This technique is the root of virtually all audio signal-processing algorithms. However, in most DSP applications, this frequency decomposition consumes vast amounts of the processor's resources and requires high power. Since the C^4s operate at very low power and in continuous time, there are significant advantages in using the analog version of this frequency decomposition. This bank of C^4s has already been shown to be useful in speech recognition systems [67, 68], noise suppression algorithms [69], and in making biometric models of the human cochlea [70].

APPENDIX A

LIST OF CHIPS FABRICATED

All the chips were fabricated in the $0.5\mu m$ CMOS technology. The following chips have been fabricated:

Vector–matrix multiplier	
Chip no.	**Description**
T16W_CR	Vector–matrix multiplier with no cascodes
T16W_CR	Vector–matrix multiplier with cascodes
Continuous-time filters	
Chip no.	**Description**
T21R_EF	Characterization chip for fully–differential OTAs
T21R_DU	Voltage–mode 2^{nd}-order sections based on the FG-OTAs
T21S_BC	Voltage–mode 6^{th}-order $G_m - C$ bandpass filter
T26Y_BG	Voltage–mode 2^{nd}-order $G_m - C$ bandpass sections based on C4s
T2AK_AU	Voltage–mode 6^{th}-order $G_m - C$ bandpass sections based on C4s
T29U_AP	Current–mode 2^{nd}-order log–domain bandpass sections
T29U_AP	Current–mode 6^{th}-order log–domain bandpass sections
Modulator/Demodulator system	
Chip no.	**Description**
T21R_EF	Chip with bldg.-blocks such as shift-register, mixers, decoders
T21R_DU	A Mod/Demod chip with an array of 64x8 FG elements

REFERENCES

[1] W. Shao, J. Xie, and G. Wang;, "Structure and implementation of smart antennas based on sortware radio," in *IEEE International Conference on Man and Cybernatics*, pp. 1938–1943, 2003.

[2] B. Porat, ed., *A Course in Digital Signal Processing*. New York: John Wiley and Sons, Inc., 1997.

[3] T. Hall, C. Twigg, P. Hasler, and D. Anderson, "Application performance of elements in a floating–gate fpaa," in *Proceedings of the International Symposium on Circuits and Systems*, vol. II, (Vancouver), pp. 589–592, 2004.

[4] R. R. Harrison, J. A. Bragg, P. Hasler, B. A. Minch, and S. Deweerth, "A CMOS programmable analog memory cell array using floating-gate circuits," *IEEE Transactions on Circuits and Systems II*, vol. 48, pp. 4–11, Jan. 2001.

[5] F. Adil and P. Hasler, "Offset removal from floating gate differential amplifiers and mixers," in *Midwest Circuits and Systems*, (Tulsa, OK), Aug. 2002.

[6] M. Kucic, A. Low, P. Hasler, and J. Neff, "Programmable continuous-time floating-gate fourier processor," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 48, pp. 90–99, January 2001.

[7] P. Smith, M. Kucic, R. Ellis, P. Hasler, and D. V. Anderson, "Cepstrum frequency encoding in analog floating-gate circuitry," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, vol. IV, (Phoenix, AZ), pp. 671–674, May 2002.

[8] P. Hasler, B. A. Minch, and C. Diorio, "Adaptive circuits using pFET floating-gate devices," in *Proceedings of the 20th Anniversary Conference on Advanced Research in VLSI*, (Atlanta, GA), pp. 215–229, March 1999.

[9] P. Hasler, B. A. Minch, and C. Diorio, "An autozeroing floating-gate amplifier," *IEEE Transactions on Circuits and Systems II*, vol. 48, pp. 74–82, Jan. 2001.

[10] P. Hasler and J. Dugger, "Correlation learning rule in floating-gate pFET synapses," *IEEE Transactions on Circuits and Systems II*, vol. 48, pp. 65–73, Jan. 2001.

[11] M. Kucic, J. Dugger, P. Hasler, and D. Anderson, "Programmable and adaptive analog filters using arrays of floating-gate circuits," in *Proceedings of the 21st Conference on Advanced Research in VLSI*, (Atlanta, GA), pp. 148 –162, March 2001.

[12] B. A. Minch, C. Diorio, P. Hasler, and C. A. Mead, "Translinear circuits using subthreshold floating-gate MOS transistors," *Analog Integrated Circuits and Signal Processing*, vol. 9, no. 2, pp. 167–179, 1996.

[13] B. A. Minch, P. Hasler, and C. Diorio, "Multiple-input translinear element networks," *IEEE Transactions on Circuits and Systems II*, vol. 48, pp. 20–28, Jan. 2001.

[14] T. Shibata and T. Ohmi, "A functional MOS transistor featuring gate-level weighted sum and threshold operations," *IEEE Transactions on Electron Devices*, vol. 39, no. 6, pp. 1444–1455, 1992.

[15] K. Yang and A. G. Andreou, "A multiple input differential amplifier based on charge sharing on a sloating-gate MOSFET," *Journal of Analog Integrated Circuits and Signal Processing*, vol. 6, no. 3, pp. 197–208, 1994.

[16] M. Lenzlinger and E. Snow, "Fowler–Nordheim tunneling into thermally grown SiO_2," *Journal of Applied Physics*, vol. 40, no. 1, pp. 278–283, 1969.

[17] 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.," *trcasII*, submitted for publication.

[18] C. Mead, "Scaling of MOS technology to submicrometer feature sizes," *Journal of VLSI Signal Processing*, vol. 8, pp. 9–25, 1994.

[19] A. Aslam-Siddiqi, W. Brockherde, and B. Hosticka, "A 16 x 16 nonvolatile programmable analog vector-matrix multiplier," *IEEE Journal of Solid-State Circuits*, vol. 33, pp. 1502–1509, Oct. 1998.

[20] G. Serrano, P. Smith, H. Lo, R. Chawla, T. Hall, C. Twigg, and P. Hasler, "Automated Rapid Programming of Large Arrays of Floating-gate Elements," in *International Symposium on Circuits and Systems*, 2004.

[21] Y. Tsividis, ed., *Operation and Modeling of the Mos Transistor*. Oxford University Press, 2003.

[22] M. Kucic, A. Low, P. Hasler, and J. Neff, "A programmable continuous-time floating-gate fourier processor," *IEEE Transactions on Circuits and Systems II*, vol. 48, no. 1, 2001.

[23] M. Kucic, P. Hasler, J. Dugger, and D. V. Anderson, "Programmable and adaptive analog filters using arrays of floating-gate circuits," in *2001 Conference on Advanced Research in VLSI* (E. Brunvand and C. Myers, eds.), pp. 148–162, IEEE Computer Society, March 2001.

[24] B. Razavi, ed., *CMOS Analog IC design*. Prentice Hall, 2001.

[25] C. Yee and A. Buchwald, "A sampled-data switched-current analog 16-tap fir filter with digitally programmable coefficients in 0.8μm cmos," *IEEE Int. Solid-State Circuits Conference*, vol. 33, pp. 54–54, Feb 1997.

[26] H. Fujishima, Y. Takemoto, T. Onoye, and I. Shirakawa, "An architecture of a matrix-vector multiplier dedicated to video decoding and three-dimensional graphics," *IEEE Transactions on Circuits and Systems II*, vol. 9, pp. 306–314, Mar. 1999.

[27] R. Genov and G. Cauwenberghs, "Charge-mode parallel architecture for vector-matrix multiplication," *IEEE Transactions on Circuits and Systems II*, vol. 48, pp. 930–936, Oct. 2001.

[28] N. Khachab and M. Ismail, "A 16 x 16 nonvolatile programmable analog vector-matrix multiplier," *Electron Letters*, vol. 33, pp. 1502–1509, Oct. 1998.

[29] H. Mehrvarz and C. Kwok, "A novel multi-input floating-gate mos four quadrant analog multiplier," *IEEE Journal of Solid-State Circuits*, vol. 31, pp. 1123–1131, Aug. 1996.

[30] R. Genov and G. Cauwenberghs, "Charge–mode parallel architecture for vector–matrix multiplication," *trcasII*, vol. 48, pp. 930–936, Oct 2001.

[31] F. Kub, K. Moon, I. Mack, and F. Long, "Programmable analog vector-matrix multipliers," *IEEE Journal of Solid-State Circuits*, vol. 25, pp. 207–214, Feb. 1990.

[32] H. Song and C. Kim, "An nmos four-quadrant analog multiplier using simple two-input squaring circuits with source followers," *IEEE Journal of Solid-State Circuits*, vol. 25, pp. 841–848, June 1990.

[33] A. Bandyopadhyay and P. Hasler, "A fully programmable cmos block matrix transform imager architecture," in *Custom integrated circuits conference*, 2003.

[34] S. Berner and P. Leon;, "Parallel digital architectures for high-speed adaptive dsss receivers," in *IEEE International Conference on Man and Cybernatics*, pp. 1298–1302, 2000.

[35] Y.-P. Lin and S.-M. Phoong, "Analog representation and digital implementation of ofdm systems," in *Proceedings of the International Symposium on Circuits and Systems*, vol. 4, pp. 9–12, May 2003.

[36] J. Vankka, J. Ketola, J. Sommarek, O. Vaananen, M. Kosunen, and K. Halonen, "A gsm/edge/wcdma modulator with on-chip d/a converter for base stations," *IEEE Transactions on Circuits and Systems II*, vol. 49, pp. 645 – 655, October 2002.

[37] H. V. Tran, T. Blyth, D. Sowards, L. Engh, and B. Nataraj, "A 2.5v, 256-level non-volatile analog storage device using eeprom technology," in *Proceedings of ISSCC,*, pp. 270–271, 1996.

[38] F. Rashidi, M. Rashidi, and A. Hashemi-Hosseini, "Design of a modulator and demodulator ssb by using dsp processor tms320c50 for plc systems," in *IECON*, vol. 2, pp. 1775–1779, Nov 2003.

[39] A. Batra, J. Balakrishnan, and A. Dabak, "Multiband ofdm: Why it wins for uwb," in *CommsDesign article*, pp. 1–7, June 2003.

[40] B. Razavi, ed., *RF Microelectronics*. Prentice Hall, 2000.

[41] A. Ghazel, M. Zhili, and N. Youssef;, "Optimized dsp implementation of gmsk software modem for gsm transceiver," in *Proceedings of Vehicular Technology Conference*, pp. 2573 – 2577, 2000.

[42] D. J. and Ken Martin, ed., *MOS Design*. Prentice Hall, 2001.

[43] P. Kallam, E. Sanchez-Sinencio, and A. Karsilayan, "An enhanced adaptive q-tuning scheme for a 100-mhz fully symmetric ota-based bandpass filter," *IEEE Journal of Solid-State Circuits*, vol. 38, pp. 585 – 593, April 2003.

[44] T. Pham and P. Allen, "A highly accurate step-response-based successive-approximation frequency tuning scheme for high-q continuous-time bandpass filters," *IEEE Transactions on Circuits and Systems I*, vol. 50, pp. 221 – 227, May 2003.

[45] C. Salthouse and R. Sarpeshkar, "A practical micropower programmable band-pass filter for use in bionic ears," *IEEE Journal of Solid-State Circuits*, vol. 38, pp. 63 – 70, January 2003.

[46] E. Sanchez-Sinencio and J. Silva-Martinez, "CMOS Transconductance Amplifiers, Architectures and Active Filters:A Tutorial," in *IEEE Proceedings-Circuits, Devices and Systems*, vol. 147, pp. 3–12, Feb 2000.

[47] A. Veeravalli, E. Sanchez-Sinencio, and J. Silva-Martinez, "Transconductance Amplifier Structures with very small Transconductances: A Comparative Design Approach.," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 6, pp. 770–775, 2002.

[48] H. Elwan and M. Ismail, "A CMOS Digitally Programmable Class AB OTA Circuit," *IEEE Transactions on Circuits and Systems-II:Analog and Digital Signal Processing*, vol. 347, pp. 1551–1556, Dec 2000.

[49] F. Adil, G. Serrano, and P. Hasler, "Offset removal using Floating-Gate circuits for Mixed-Signal Systems," in *Proceedings of the IEEE Southwest Symposium on Mixed-Signal Design*, Feb 2003.

[50] R. Chawla, G. Serrano, D. Allen, A. Periera, and P. Hasler, "Fully differential floating-gate programmable otas with novel common-mode feedback," in *Proceedings of the International Symposium on Circuits and Systems*, May 2004.

[51] P. Smith, M. Kucic, R. Ellis, P. Hasler, and D. V. Anderson, "Mel-frequency cepstrum encoding in analog floating-gate circuitry," in *Proceedings of the International Symposium on Circuits and Systems*, vol. IV, (Phoenix, AZ), pp. 671–674, May 2002.

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

[53] P. Hasler, M. Kucic, and B. A. Minch, "A transistor-only circuit model of the autozeroing floating-gate amplifier," in *Midwest Conference on Circuits and Systems*, (Las Cruces, NM), 1999.

[54] B. A. Minch, P. Hasler, and C. Diorio, "The Multiple-Input Translinear Element: A Versatile Circuit Element," in *International Symposium on Circuits and Systems*, vol. 1, pp. 527–530, 1998.

[55] B. A. Minch, "Multiple-input translinear-element log-domain filters," *IEEE Transactions on Circuits and Systems II*, vol. 48, pp. 29–36, Jan. 2001.

[56] D. R. Frey, "Exponential state space filters: A generic current-mode design strategy," in *International Symposium on Circuits and Systems*, vol. 43, pp. 34–42.

[57] B. A. Minch, "Multiple-Input Translinear Element Log-Domain Filters," in *International Symposium on Circuits and Systems*, vol. 48, pp. 29–36, 2001.

[58] R.N.Caffin, "On Some Aspects of the High-Frequency Performance of Operational Amplifiers and Current-to-Voltage Converters," *IEEE Journal of Solid-State Circuits*, vol. 10, pp. 503–505, Dec. 1975.

[59] E. B.Nauta and W.Kruiskamp, "A CMOS Triode Transconductor," *Proceedings of the International Symposium on Circuits and Systems*, vol. 4, pp. 2232–2235, June 1991.

[60] A. Nedungadi and T.R.Viswanathan, "Design of Linear Transconductance Elements," *IEEE Transactions on Circuits and Systems*, vol. 31, pp. 891–894, Oct. 1984.

[61] K.C.Kuo and A.Leuciuc, "A Novel Linear Tunable MOS Transconductance," *IEEE Journal of Solid-State Circuits*, vol. 36, pp. 147–151, Jan. 2001.

[62] A.-M. R.Shukla, J.Ramirez-Angulo and R.G.Carvajal, "A Low Voltge Rail to Rail V-I Conversion Scheme for Applications in Current Mode A/D converters," *Proceedings of the International Symposium on Circuits and Systems*, vol. 1, pp. 916–919, May 2004.

[63] C.Wang and J.Wang, "Design of Linear Transimpedance Amplifiers," pp. 232–235, Oct. 2001.

[64] R. Chawla, A. Bandyopadhyay, V. Srinivasan, and P. Hasler, "A 531 nw/mhz, 128x32 currentmode programmable analog vectormatrix multiplier with over 2 decades of linearity," in *Custom Integrated Circuits Conference*, 2004.

[65] R. Chawla, H.-J. Lo, A. Basu, P. Hasler, and B. A. Minch, "A fully programmable log-domain bandpass filter using multilple-input translinear elements," in *Proceedings of the International Symposium on Circuits and Systems*, May 2004.

[66] A. Bandyopadhyay, *Matrix Transform Imager Architecture for OnChip Low-Power Image Processing*. PhD thesis, Georgia Institute of Technology, 2004.

[67] P. D. Smith, M. Kucic, R. Ellis, P. Hasler, and D. V. Anderson, "Mel-frequency cepstrum encoding in analog floating-gate circuitry," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, (Scottsdale, AZ), May 2002.

[68] T. Massengill, D. Wilson, P. Hasler, and D. Graham, "Empirical comparison of analog and digital auditory preprocessing for automatic speech recognition," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, (Scottsdale, AZ), May 2002.

[69] R. Ellis, H. Yoo, D. Graham, P. Hasler, and D. Anderson, "A continuous-time speech enhancement front-end for microphone inputs," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, (Scottsdale, AZ), May 2002.

[70] D. Graham and P. Hasler, "Capacitively-coupled current conveyer second-order sections for continuous-time bandpass filtering and cochlea modeling," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, (Scottsdale, AZ), May 2002.