APPLICATION OF FLOATING-GATE TRANSISTORS IN FIELD PROGRAMMABLE ANALOG ARRAYS

A Thesis
Presented to
The Academic Faculty

By

Jordan D. Gray

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
in
Electrical and Computer Engineering

School of Electrical and Computer Engineering
Georgia Institute of Technology
December 2005

APPLICATION OF FLOATING-GATE TRANSISTORS IN FIELD PROGRAMMABLE ANALOG ARRAYS

Approved by:

Dr. Paul Hasler, Advisor
Assoc. Professor, School of ECE
Georgia Institute of Technology

Dr. David Anderson
Assoc. Professor, School of ECE
Georgia Institute of Technology

Dr. Farrokh Ayazi
Assoc. Professor, School of ECE
Georgia Institute of Technology

Date Approved: November 22, 2005

ACKNOWLEDGMENT

I rarely get an opportunity to formally acknowledge those who have put me on this path. That being the case, I would like to take the opportunity to thank my parents. My father, Harvey Gray, gave me the analytical eye I cherish so much while my mother, Leslie Simmons, showed me how to use it. It is interesting, I suppose, that the only place their marriage ever made sense was with respect to the burden of my insight. Thank you both!

Ashley Mastin, my love, you penetrate my mind and twist me in knots. Without you my research would take twice as long—you are the reason I rush home every evening. I cannot acknowledge you enough.

To all of my lab mates, friends, and colleagues, you have made me who I am. Chris Twigg and Tyson Hall, you have opened the door to the FPAA for me. Dave Abramson and Shyam Subramanian, without you there would be no MITE FPAA. And Brad Minch, without you there would be no MITEs at all! Venkatesh Srinivasan and Sheng-Yu Peng, you two are my role models. I hope I have the opportunity to mentor by example in the same way you two have done for me. Ryan Robucci. What can I say? You dull my edges and sharpen my mind. Paul Hasler, my advisor, you are the master visionary. You accepted me with nothing but my enthusiasm.

Finally, I must acknowledge the universe. It works. Thank you.

TABLE OF CONTENTS

LIST OF FIGURES

SUMMARY

Floating-gate transistors similar to those used in FLASH and EEPROM can be used to build reconfigurable analog arrays. The charge on the floating gate can be modified to pass or block a signal in a cross-bar switch matrix, or it can be finely tuned to eliminate a threshold difference across a chip or set a bias. By using such a compact and versatile reconfigurable analog memory element, the number of analog circuit components included on an integrated circuit that is field-programmable is significantly higher. As a result, large-scale FPAAs can be built with the same impact on analog design that FPGAs have had on digital design. In my research, I investigate the areas floating-gate transistors can be used to impact FPAA design and implementation.

An FPAA can be broken up into two basic components, elements of connection and elements of computation. With respect to connection, I show that a floating-gate switch can be used in a cross-bar matrix in place of a transmission gate resulting in less parasitic capacitance and a more linear resistance for the same size transistor. I illuminate the programming issues relating to injecting a floating-gate for use as a switch, including the drain selection circuitry and rogue injection due to gate induced drain leakage. With respect to computation, I explain how a Multiple-Input Translinear Element, or MITE, can be augmented to fit in an FPAA framework. I also discuss two different MITE implementations compatible with CMOS technology, a subthreshold MOS design and a BJT MITE that uses a lateral BJT.

Beyond FPAA components, I present two alternative FPAA systems. The first is a general purpose reconfigurable analog system that uses standard analog design components that have been augmented with floating-gates. The second FPAA is built upon MITE circuits, and is focused on supporting direct system synthesis. I conclude with a discussion of a future large-scale MITE FPAA.

CHAPTER 1

INTRODUCTION

The design cycle for a traditional analog system requires design, simulation, implementation, and testing. In the case of IC analog systems, the implementation requires a physical description, i.e. VLSI, and fabrication. Unlike digital system design where system aspects can be verified on an FPGA, the physical implementation of an analog system can only be verified once fabrication is complete. This makes analog system design more time consuming and thus more expensive. Meanwhile, a resurgence of analog design is occurring as a means for satisfying the power-efficiency requirements of embedded systems. A reasonable avenue for bridging the implementation and test gap between digital and analog design is the introduction of an analog equivalent of an FPGA, a field programmable analog array (FPAA).

One of the major stumbling blocks in implementing an FPAA is being able to provide the same flexibility FPGAs provide to digital design. However, the 40 year old floating-gate technology used to build EEPROMs and FLASH memories represents a possible solution. Rather than using the charge on a floating-gate transistor as a logical 1 or 0, the charge can be used to set a particular corner frequency or switch resistance.

What follows is a look at how floating-gate transistors can be applied to FPAAs. The first chapter discusses the floating-gate concept, how the charge on such a device is programmed, and an attempt to qualify its charge retention. The second chapter is about the design and implementation of a floating-gate switch matrix. The third chapter covers floating-gate augmented computational elements. The chapter is strongly slanted towards a class of floating-gate elements that utilize the translinear principal to implement. The fourth chapter brings the aforementioned concepts together into a discussion of two FPAA systems. The fifth chapter concludes the document. It explores the impact of the work herein, establishes the role the author played in the research presented, and looks towards the future of FPAA design.

CHAPTER 2

FLOATING-GATE ELEMENTS

First formally conceived in 1967, a floating gate transistor is named as such because of the electrically isolated material that forms the gate of the transistor. As a methodology, it represents a means for implementing a non-volatile memory element in silicon CMOS technology. The floating-gate transistor is a critical element of modern micro-scale electrical circuitry as it sits at the core of FLASH memory. And though floating gates are primarily used as a storage mechanism for digital systems, there has been a trend of research and development for floating gates as an analog circuit element over the last 15 years [23, 12, 25, 22, 17, 10, 21]. By understanding the I-V relationship and charge storage issues of the floating-gate transistor, it can be used effectively to enhance analog circuit design and implementation.

2.1 Device characteristics

A floating-gate transistor in its simplest form is a standard MOS transistor with a capacitor in place of a gate contact. The device shown in Figure 1 is an example of typical floating-gate. Multiple coupling capacitors are often used in designing floating-gate transistors. The relationship between the terminal voltages and drain current of the two-input floating-gate

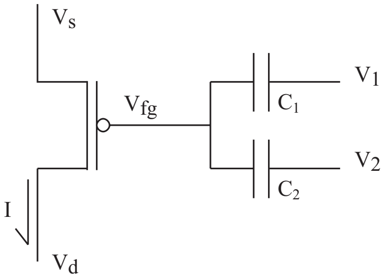


Figure 1. A basic floating-gate schematic with two coupling capacitors.

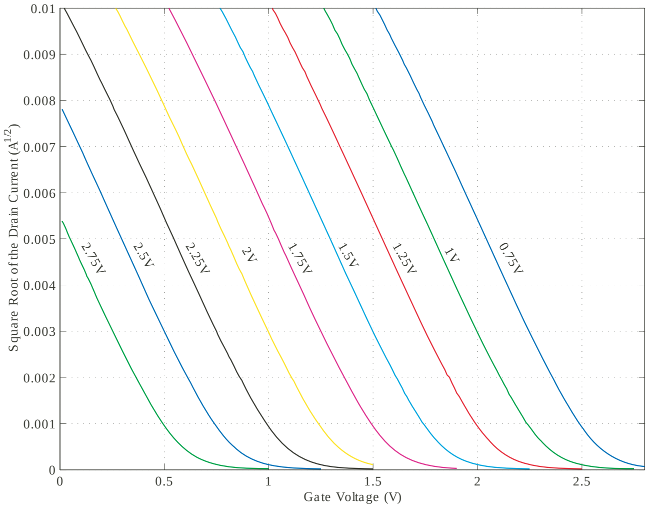


Figure 2. A floating-gate transistor programmed to different threshold values. The effective threshold voltage is given for each curve.

transistor, assuming saturated sub-threshold operation, is given by the following equation

$$I = I_s e^{\frac{(V_{DD}-V_S)-\kappa\left(V_{DD}-V_{fg}\right)}{U_T}} e^{\frac{V_D}{V_A}} \tag{1}$$

where the floating-gate voltage is formulated by the following:

$$V_{fg} = \frac{1}{C_T}\left(C_1V_1 + C_2V_2 + C_{GS}V_S + C_{GD}V_D + Q\right) \tag{2}$$

and C_T is given by

$$C_T = C_1 + C_2 + C_{GS} + C_{GD} \tag{3}$$

There are at least two important implications of Equation 2: the gate voltage is a function of the charge stored on it, and the gate voltage is a function of any other voltage capacitively coupled to the gate.

Because the gate voltage is a function of the charge stored on the floating-gate, the I-V curve of the transistor can be shifted to a particular, desirable point. Illustrated in Figure 2 are a series of gate sweeps for a floating-gate device with different amounts of charge stored. The result is a single transistor with a wide array of possible effective threshold

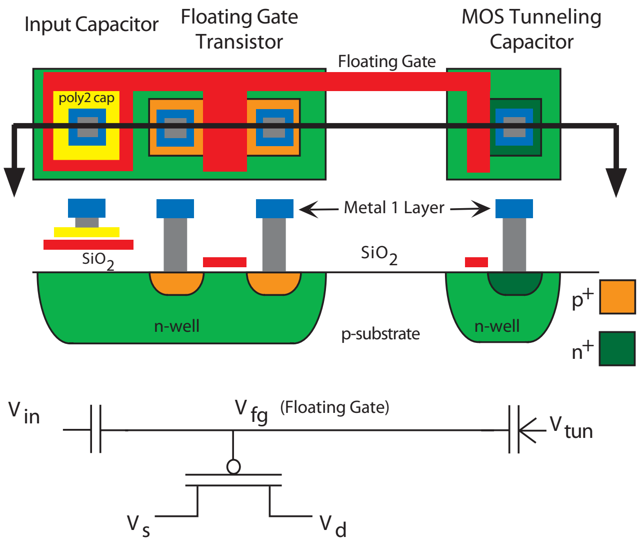


Figure 3. The layout for a floating-gate transistor. The MOS capacitor is used for charge removal while the drain of the transistor is where charge addition occurs. Both issues are covered in Section 2.2.

values, with a threshold given by:

$$V_{th} = V'_{th} + \frac{Q}{C_T} \tag{4}$$

where V'_{th} is the threshold of the same transistor without a floating gate. The implication is that for DC conditions, the current through a floating-gate transistor can be set as precisely as the charge on the floating-gate can be controlled.

The current through a floating-gate transistor under the condition that its terminals are not at fixed potentials is less obvious because of the contribution to the effective gate voltage from potentials though capacitive coupling. For instance, the Early Effect of a floating-gate transistor is usually dominated by capacitive coupling rather than the length of the transistor. The effect of the drain coupling can engineered to a nominal factor by decreasing the overlap capacitance or increasing the value of C_T.

The layout for a floating-gate transistor is shown in Figure 3. The MOS capacitor is used for charge removal while the drain of the transistor is where charge addition occurs. Both issues are covered in Section 2.2.

2.2 Floating-gate charge movement techniques

As discussed previously, the I-V characteristic of a floating-gate transistor can be set as accurately as charge movement techniques allow. There are currently two such techniques currently employed: electron tunneling and hot-carrier injection.

2.2.1 Tunneling

Electron tunneling is the process by which an electron passes through a barrier rather than traversing the conduction band associated with that barrier. In the case of direct band-to-band tunneling, electrons may pass though a barrier without any assistance. The likelihood of direct band-to-band tunneling is related to the thickness of the barrier. Obviously, oxides that demonstrate appreciable levels of direct band-to-band tunneling are unsuitable for building an electrically isolated gate.

In the case where the oxide thickness is not so thin that spontaneous tunneling dominates, a field-assisted mechanism called Fowler-Nordheim tunneling can be used to modify charge stored on a floating material. Illustrated in Figure 4, electrons on the floating-gate are trapped by the barrier imposed by the SiO_2. By lowering the voltage of the silicon, the energy bands bend in such a way that the electrons see a thinner, triangular barrier. As a result, electrons tunnel through the material. The process is sometimes referred to as a tunneling diode since the current can only move in one direction. As a result, the system is in negative-feedback, constantly slowing the rate of tunneling. As current flows through the tunneling diode, the loss of electrons on the polysilicon results in a positive change in potential. The net result is a widening of the triangular barrier and a decrease in tunneling current.

2.2.2 Injection

Hot-carrier injection is the process by which a carrier is excited to the point that it can surmount an interface barrier and enter the region of the associated barrier material's conduction band. Common forms of hot-carrier injection include UV light exposure, channel

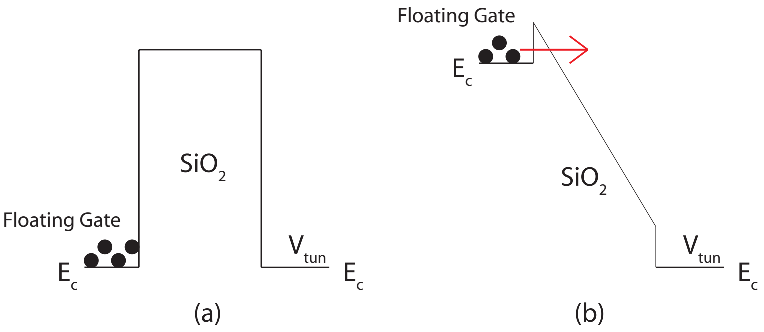


Figure 4. An illustration of Fowler-Nordheim tunneling. (a) Initially, the SiO_2 inhibits tunneling. (b) By creating a significant voltage difference across the barrier, the conduction band bends until carriers can tunnel though the narrow triangular region.

hot-carrier injection, and drain avalanche hot-carrier (DAHC) injection [3][30].

In EPROM memory structures, UV light is typically used to erase the device before it is programmed. The dominant charge movement mechanism is injection. UV light generates carriers in the silicon and imparts them with enough energy to promote the carriers into the conduction band of the dielectric that isolates the floating-gate of the EPROM.

Channel hot-carrier injection relies on creating a high gate and drain potential relative to the source. A minority carrier then leaves the source with the intention of going to the drain. As result of the high field, the carrier gains enough energy to surmount the interface barrier and ends up in the conduction band of the barrier rather than making it to the drain.

Then there is drain avalanche hot-carrier (DAHC) injection. There is still a high gate and drain potential relative to the source. In this case, a minority carrier makes it all the way from the source to the drain. However, the field has provided enough energy to cause the carrier to impact ionize the silicon in the drain region. The carriers at the drian avalanche and majority carriers, who would have otherwise traveled down into the bulk, have enough energy and are presented with the right field necessary to surmount the interface barrier and enter the conduction band of the material responsible for the barrier. An illustration of the process is provided in Figure 5. A pMOS transistor is shown since it is the floating-gate device type used in this thesis. The reasoning behind the use of a pMOS is covered in section 2.3. One of the fundamental assumptions in relying on DAHC is that a channel can

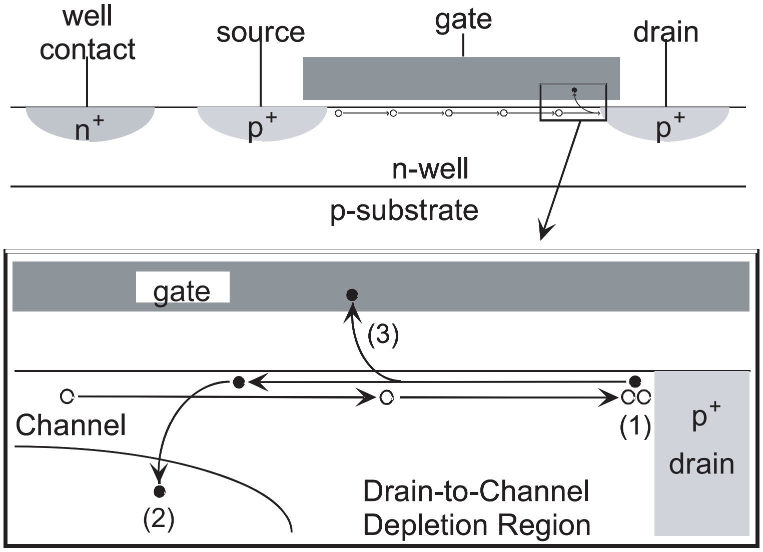


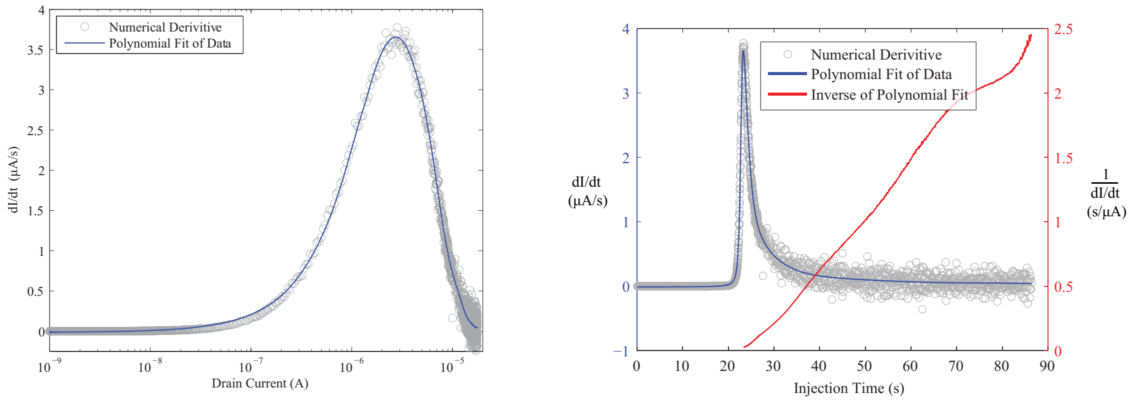
Figure 5. Illustration of DAHC injection. (1) The minority carrier impact ionizes the drain region, creating an electron-hole pair. (2) Majority carriers are swept out into the bulk. (3) Because of the gate-to-bulk field, a portion of the high-energy majority carriers inject into the conduction band of the barrier and enter onto the floating-gate.

be created, since a channel is necessary to accelerate the minority carrier from the source to the drain.

2.3 Programming pMOS transistors

While nMOS transistors are the dominant device choice for FLASH and EEPROM non-volatile digital memory structures, special processing steps are generally required to sustain reliable, consistent operation. On the other hand, pMOS transistors with sufficiently large gate oxides available in standard CMOS processing are well suited for direct floating-gate implementation [11]. As a result, the following programming techniques are expressed for pMOS floating gates on standard CMOS processes exclusively.

The FPAAs discussed in this document rely on a large number of floating-gate transistors. While the programming methods discussed in the following subsections relate only to single transistors, the framework for interacting with arrays of floating-gate transistors are covered in Section 3.2.1.



(a) (b)

Figure 6. This is experimental data for the time-derivative of current during injection. The V_{SD} was kept small enough that measurement on the order of seconds was possible. (a) Peak injection occurs relative to a particular current level. (b) At a time after peak injection, the rate of current change falls off with approximately a $\frac{1}{X}$ dependence.

2.3.1 Removing electrons

In order to create the triangular barrier necessary to tunnel electrons off of the floating gate, extremely high voltages must be used. In many cases, the voltage for tunneling exceeds the breakdown voltage for the active-to-bulk PN junctions in a process, making on-chip instrumentation difficult. In addition, charge movement through Folwer-Nordheim tunneling is less well characterized for floating-gates than injection. As a result, it is more convenient to use tunneling as a global erase. Each floating gate has a tunneling capacitor that consists of a MOS-cap as shown in Figure 3. A MOS capacitor is used for tunneling since it is the highest quality oxide available in a standard CMOS process.

2.3.2 Adding electrons

Channel hot-carrier injection is a common technique for adding electrons to floating-gate nFETs because an electron is the minority carrier to be injected. In an pMOS device, the majority carrier is an electron, so one would expect reasonable current densities resulting from DAHC injection. As a result, the following techniques relate to for DAHC injection.

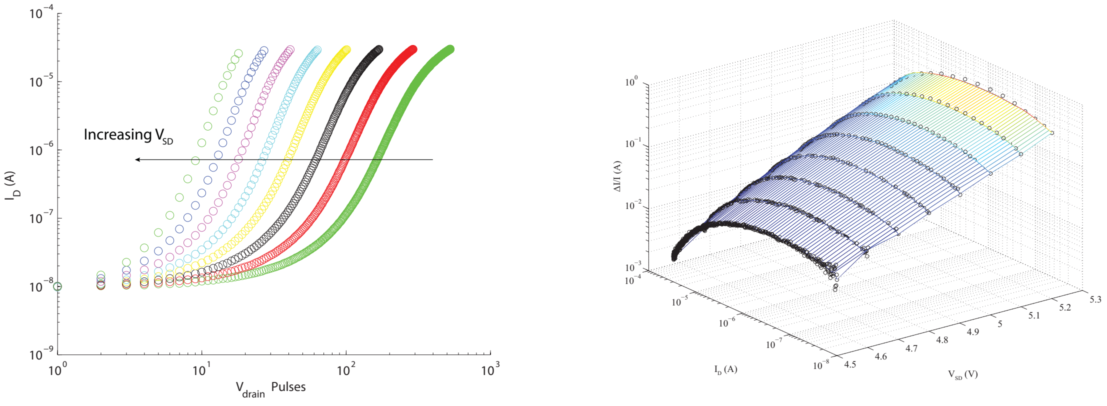
2.3.2.1 Gate-sweep injection

Gate-sweep injection is used when exact currently levels are not important and results in high levels of injection. The need for a gate sweep results from the observation of the instantaneous rate of change of the drain current, shown in Figure 6. The instantaneous change in the current relates to the efficiency of injection. Figure 6a illustrates that peak injection is related to a particular current level which corresponds to a particular effective gate voltage. Moreover, by injecting electrons onto a floating-gate the effective voltage is constantly changing. In order to counteract the negative feedback from the accumulation of charge, the gate must be constantly moved in order to maintain injection.

Often there is so much charge on the floating gate that it is not possible to inject at maximum efficiency. As a result, a linear gate sweep is not necessarily the best choice. As illustrated in Figure 6b, beyond the maximum injection efficiency the rate of injection falls off at approximately $\frac{1}{X}$. When injecting a device that cannot be brought back to peak injection current levels, it is necessary to spend a longer time injecting. In particular, it is often beneficial to use a gate sweep with a logarithmic characteristic. The logarithmic curve allows for a higher density of points at higher voltages, counteracting the reduced injection efficiency.

2.3.2.2 Drain-pulse injection

Drain-pulse injection is a characterization intensive programming methodology and results in very efficient, accurate programming. It works by injecting a floating-gate transistor in short bursts or pulses, and is more completely described in [24]. The illustration in Figure 7a is the first step in the process. A transistor has been injected for over a wide range of V_{SD} voltages. Because pulses are used, a derivative is not possible because of parasitic effects relating to the rising and falling edge of the pulse. Instead, the percentage change is used as a means for evaluating the injection efficiency. The data, plotted as black circles in Figure 7b, relates a particular drain current and V_{SD} to a percentage change in the floating-gate transistor current. The data is curve-fit resulting in the surface of Figure 7b. It represents



(a) (b)

Figure 7. (a) The raw data used to characterize drain-pulse injection. (b) The data is repotted as $\frac{\Delta I}{I}$ vs I, the black circles, and curve fit, resulting in the surface shown.

a mapping from one current level to another. When a particular floating-gate current is necessary, the mapping provides the necessary V_{SD} to reach the new current level.

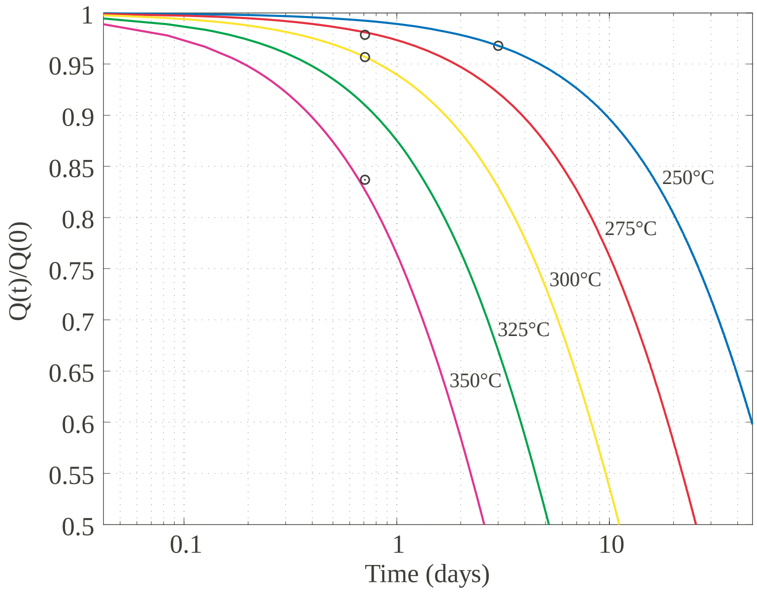
2.4 Charge retention

The quality of the circuits and systems discussed in this document are proportional to the precision that charge can be injected onto electrically isolated polysilicon. Therefore a key characteristics of a floating-gate device is its ability to retain charge for long periods of time with minimal leakage. The work in this section has been published and is used to characterize the offset variation of an amplifier over time with respect to charge loss due to a floating-gate offset removal technique [27].

Assuming a high-quality oxide, the mechanism for losing charge over time is thermionic emission [19, 4]. The amount of charge lost is a function of both temperature and time and is given by,

$$\frac{Q(t)}{Q(0)} = exp\left[-t\upsilon \cdot exp\left(\frac{-\phi_B}{kT}\right)\right] \tag{5}$$

where $Q(0)$ is the initial charge on the floating-gate, $Q(t)$ is the floating-gate charge at time t, υ is relaxation frequency of electrons in poly-silicon, ϕ_B is the $\mathrm{Si}-\mathrm{SiO_2}$ barrier potential, k is the Boltzmann's constant, and T is the temperature. For normal temperatures, the



	Programmed 10% change in current			**Programmed 50% change in current**		
Temperature	**ΔQ/Q**	**ΔV_{fg}**	**ΔI/I**	**ΔQ/Q**	**ΔV_{fg}**	**ΔI/I**
25°C	1e-3%	36.7nV	2e-4%	1e-3%	156nV	9e-4%
90°C	0.62%	16.3µV	0.06%	0.62%	65µV	0.57%
140°C	18.2%	1.8mV	1.8%	18.2%	1.92mV	10.7%

Figure 8. The plot shows the measured charge loss (○'s) plotted with an extrapolated theoretical fit (solid) for different temperatures and time. The table summarizes the percentage change in the floating-gate charge, voltage, and current over ten years for two different cases: (a) 10% **programming change from initial (b)** 50% **programming change from initial.**

amount of charge lost is typically very small and difficult to measure. By increasing the temperature, the thermionic emission can be increased to measurable levels.

Knowing the exact charge on a floating-gate is not straightforward. An easier way to approach $\frac{Q(t)}{Q(0)}$ is to take advantage of a ratio of threshold voltages before and after programming. By rearranging Equation 4,

$$\frac{Q(t)}{Q(0)} = \frac{V_{th}(t) - V'_{th}}{V_{th}(0) - V'_{th}} \tag{6}$$

The values of parameters υ and ϕ_B were estimated to be $60s^{-1}$ and $0.9eV$ using experimentally measured values of charge loss for different time periods when the devices were exposed to high temperatures ($> 250^{\circ}C$) for a prolonged period of time.

Figure 8 shows the measured floating-gate charge loss along with a theoretical extrapolated fit using the estimated model parameters. Also in Figure 8 is a summary of the percentage change in floating-gate charge between two floating-gate transistors programmed to different thresholds. Pairs of floating-gate transistors were used in order to avoid the dependence on measured charge. The two different cases were: 10% programming change from initial and 50% programming change from initial. The measured data agrees well with the theoretical prediction and the trends observed in Figure 8 have been observed across many floating-gate devices. The values in the table inset in Figure 8 have been evaluated using (5) and assuming a sub-threshold operation. No significant change can be extrapolated for programmed currents for a period of 10 years at room temperature, indicating good charge retention in floating-gate devices.

CHAPTER 3

SWITCH MATRIX

Any reconfigurable system can be partitioned into to an elements of connection and elements of computation. This chapter is about elements of connection, or more simply, switches. The choice of switch and the way it is used in a design has an overwhelming impact on the reconfigurability, size, and bandwidth of the resulting system. This chapter is broken up into two parts: the characterization of a switch and the implementation of the routing matrix.

3.1 Switch characterization

An ideal transmission element has two states, *on* and *off*. In an *on* state, the element has an infinite resistance looking into both the input and output terminals. In an *off* state, the element has zero resistance looking into both the input and output terminals. Independent of the state, the ideal transmission element exhibits zero capacitance and zero inductance. In addition, the input and output terminals are interchangeable with a fixed conductance of one. Finally, the ideal element should not consume area. Obviously, the aforementioned values are a combination of impossible and infeasible. However, they serve to illustrate the issues a circuit designer is concerned with when evaluating the quality of a transmission element. With the exception of inductance, a characteristic outside the purview of this thesis due to the frequency range and line length of concern within this document, real transmission elements will be evaluated with respect to how well they mimic an ideal transmission element.

FPAAs are most commonly use MOS transistor switches driven by digital memory[14]. Alternatives to pass-FETs include G_m-C amplifiers, 4-transistor transconductors, and current conveyors[13][20][2]. All of the listed alternatives trade area for improved switch characteristics, and still require physical memory for maintaining connectivity. Ultimately,

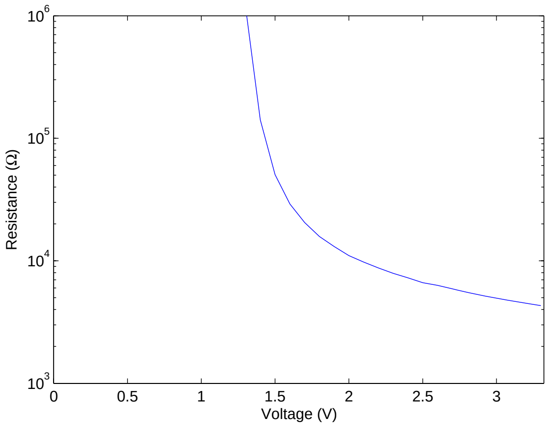


Figure 9. Experimental data for the resistance of a pass-gate. The pFET is 3.6 μm wide by 1.2 μm long and operated with a supply of 3.3 V

a floating-gate implemented as a pass-gate switch was chosen as the transmission element for the FPAAs in this thesis because of its size, value storage, and *on* resistance. The *off* resistance for all the switches discussed is higher than available ohm-meters allowed. What follows is a characterization of a pass-gate, a transmission-gate, and a floating-gate switch. This work has also been published as [6].

3.1.1 Pass-gate

The most basic form of switch is a pass-gate. Implemented in CMOS technology as a single pMOS or nMOS transistor, a pass-gate is at first a desirable choice as a switch because it is small. However, the non-linear I-V relationship of a transistor results in a less than desirable *on* resistance.

Figure 9 illustrates the most significant downside of a single transistor switch. It has a non-linear resistance that makes it unsuitable for rail-to-rail signal conveyance. The device in the figure is a pMOS transistor with a drawn width of 3.6 μm and a length of 1.2 μm. The source was swept from 25 mV to 3.3 volts with V_{SD} fixed at 25 mV while measuring channel current.

Even though a pass-gate limits the signal range it can convey, it is the most miniature switch element available in a standard CMOS process. As a result, it represents the smallest

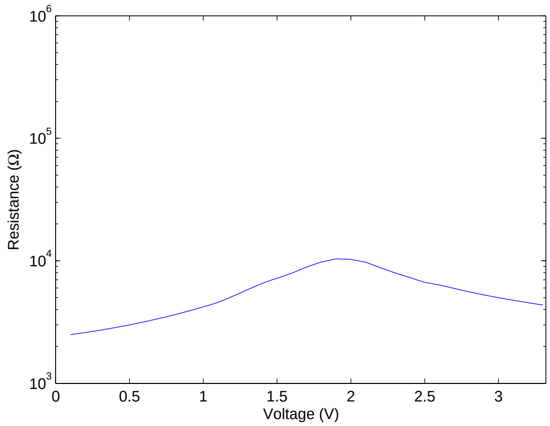


Figure 10. Experimental data for the resistance of a t-gate. The pFET is 3.6 μm wide by 1.2 μm long and operated with a supply of 3.3 V

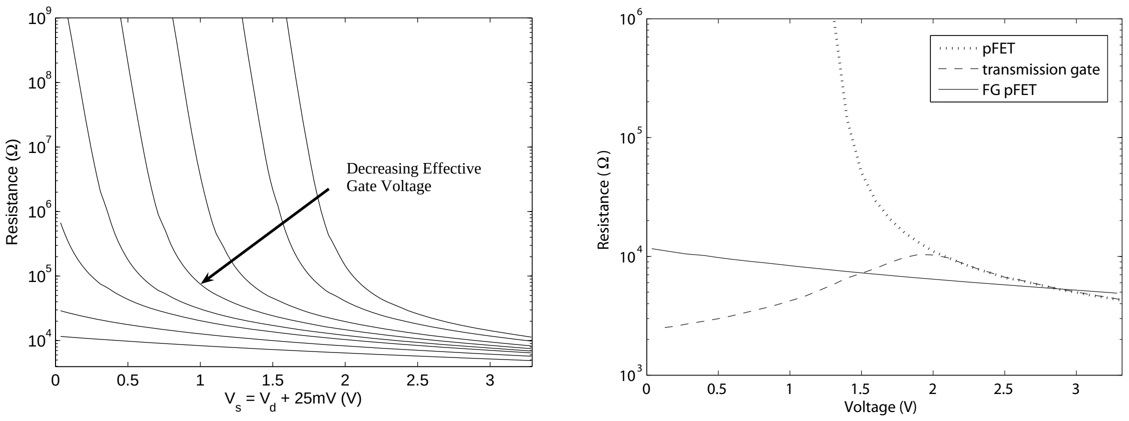
parasitic capacitance a passive switch element can contain. In systems where hundreds of switches are present on a shared transmission line, the bandwidth of the system can quickly become limited by the parasitic caps associated with the switch elements. In the case of an *off* switch, the parasitic capacitance is a combination of the active-to-bulk PN junction capacitance and the poly to active overlap capacitance. In the case of an *on* state, the contribution of parasitic capacitance to the line is the combination of the two active-to-bulk PN junctions, the channel-to-bulk, and the two poly-to-bulk overlap caps.

One final limitation of the pass-gate is the lack of switch state storage. In order to maintain the value of the switch, a memory or register is necessary. And in order to provide non-volatile storage, an EEPROM or FLASH memory is also required, increasing the total size cost of the element.

3.1.2 Transmission-gate

The transmission-gate is a common improvement to a pass-gate. Complimentary transistors are used in order to increase the signal range passible by the switch. The graph in Figure 10 demonstrates the resistance for the same voltage sweep as in section 3.1.1.

The device has an improved, more steady resistance than a pass-gate. However, it also has an increased size since it requires two transistors. As a result, the parasitic capacitance



(a) (b)

Figure 11. (a) Resistance of a floating-gate switch for different levels of injection ranging from a floating gate voltage of 0.5 V down to -3 V. (b) An empirical comparison of the on-resistance of three switch elements: a pFET, a transmission gate, and a floating-gate pFET. The floating-gate pFET, injected to a floating gate voltage of -3 V, has a comparable resistance to the t-gate.

is higher for a t-gate. It also suffers from the same lack of value storage.

3.1.3 Floating-gate switch

A floating-gate switch is simply a floating-gate transistor that is used as a switch. The switch-like nature of the transistor is set by the ability to fix the charge on the floating gate to extreme levels. In the case of an *on* switch, electrons are injected onto the floating gate until the resistance curve is relatively flat over the operating range, as shown in Figure 11a. In the case of an *off* switch, the device is tunneled until no channel forms for any coupling voltage.

Figure 11b compares the three switch types. It is clear that with an on resistance on the order of a t-gate and the parasitic cap equal to a pass-gate, a floating-gate switch is a desirable element of connection. The instrumentation for a floating-gate does have a size impact and is discussed in the following section.

3.2 Implementation

A reconfigurable routing framework for an IC can come in a number of different forms. A good starting point for consideration is the full cross-bar switch matrix. So-called because

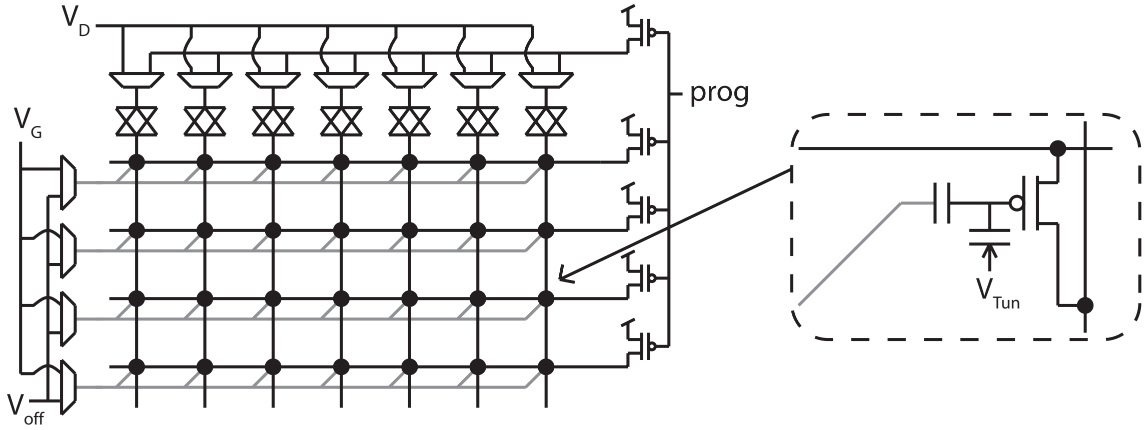


Figure 12. Switch matrix array with basic selection circuitry. Each black dot in the array is a floating gate transistor. The circuitry around the outside of the crossed-bars is necessary for programming the individual floating-gates.

it looks liked crossed bars, the routing topology affords any row to be connected to any column. Figure 12 is a representation of a 4x7 switch matrix with the basic programming circuitry included.

Each black dot in the array is a floating gate transistor. The circuitry around the outside of the crossed-bars is necessary for programming the individual floating-gates. When the pass and t-gate controls are de-asserted, rows and columns of the floating-gate transistors are used as switches and the array is said to be in *run* mode. In the case where voltages are asserted along the rows and columns of the matrix, the array is said to be in *prog* mode because variations in the asserted voltages can result in programming of the floating gates. For the purpose of visibility, the logic for setting t-gate and MUX bits is not shown. The choice of such logic is covered in Section 3.2.1. In addition, the pull-up transistors controlled by the program signal are drawn on the right side instead of the left for visibility as well. Programming circuitry is generally centralized in order to reduce the total area impact.

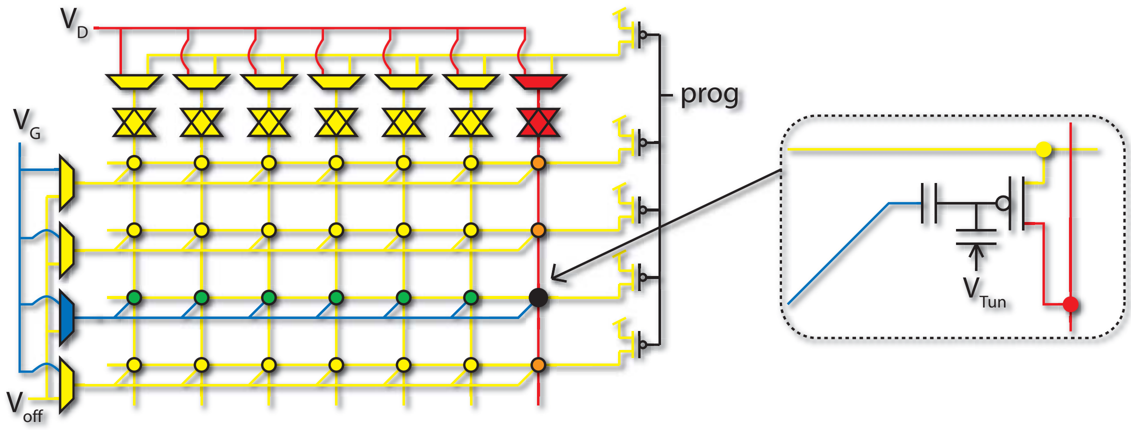


Figure 13. Floating-gate array isolation. In programming mode, all of the sources, drains, and gates are driven to V_{DD}, yellow, except a selected gate line, blue, and drain line, red. The intersection of the variable gate and drain line allow for a single floating-gate transistor to be selected. Other devices lack the V_{SG} or V_{SD} to be affected.

3.2.1 Array programming

The goal of array programming is to provide a methodology for controlling the charge on a large number of floating-gate transistors in a spatially and temporally efficient manner. The circuit topology of Figure 12 is designed with the intention of programming a single floating-gate transistor at a time with a focus on spatial efficiency. Decoders are used to select which row and column are connected to V_G and V_D respectively, and are collocated with the programming circuitry around the periphery of the array. The V_{off} voltage is used to reduce undesirable injection on rows that are unselected and is covered in Section 3.2.2.2. The t-gates along the top of the array are necessary because in *run* mode, the drains should be connected to something other than the programming circuitry.

In order to program a single floating-gate in the array, isolation of the desired device is important. In order to achieve the isolation of a single transistor only a single row and column of the array are selected. Shown as the circled transistor in Figure 13, the intersection of the selected row and column provide the source, drain, and gate potentials necessary to form a channel. This is important because, as discussed in Section 2.2.2, a channel is necessary for DAHC injection.

3.2.2 Switch isolation

When programming switches, it is not possible to maintain isolation along a column. In order to achieve the rail-to-rail low resistance, the floating-gate transistor must be injected so high that the device conducts for all V_G values. Once the device conducts for all gate voltages, it is not possible to precisely read any transistor current on the line other than the the programmed switch. As a result, it is necessary to put non-switch elements on separate drain lines.

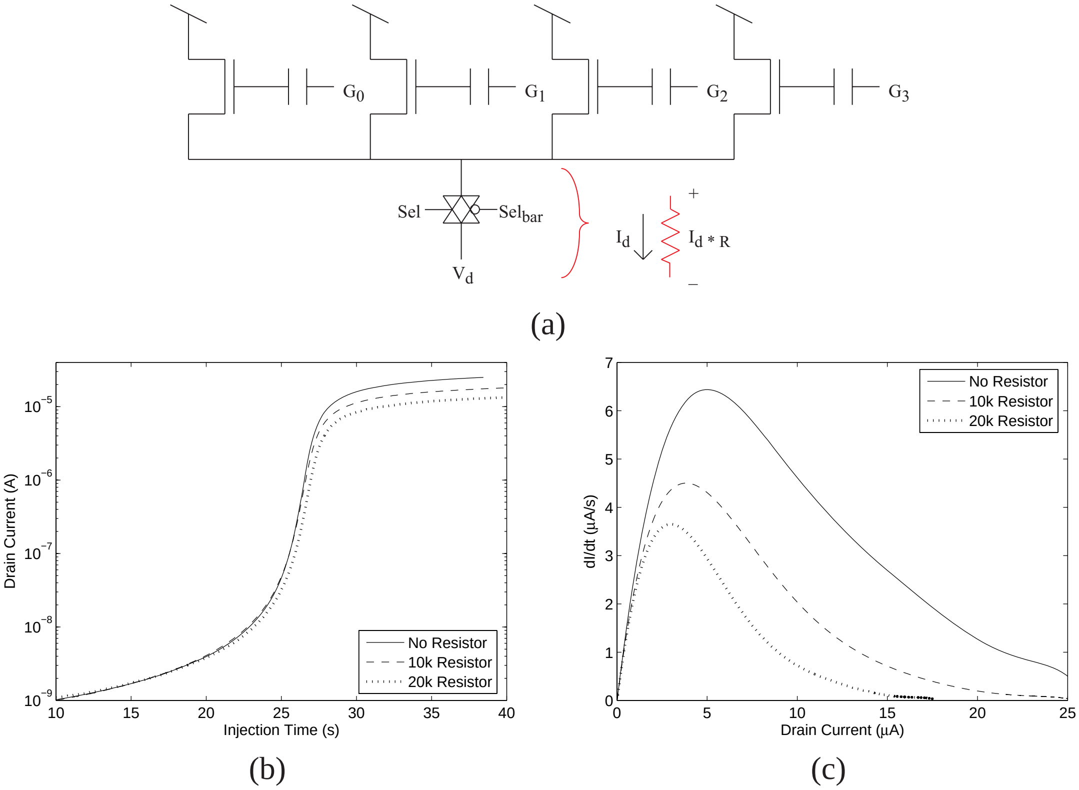
3.2.2.1 Drain selection limitations

A secondary problem of programming switches is that the current levels of an *on* switch becomes so great that a voltage drop across the drain-line t-gate is significant. The problem is illustrated schematically in Figure 14a. In order to evaluate the problem, a single, isolated floating-gate pFET was injected over a period of about 40 seconds with a V_{SD} of 5.5V. This V_{SD} was chosen to provide a measurable injection current. The device was tunnelled and then injected with a discrete 10 $k\Omega$ resistor placed between the drain node and the voltage source for the drain. The experiment was repeated for a 20 $k\Omega$ resistor. The resistors represent the possible worst case parasitic resistance due to a transmission gate. The addition of the resistors resulted in a lower drain current, implying that the injection was limited. From the injection equation[26],

$$I_{inj} = I_{inj0}(\frac{I_S}{I_{S0}})^{\alpha} e^{-V_{SD}/V_{inj}}, \tag{7}$$

it is shown that the injection is in part controlled by V_{SD}. As the current through the drain increases, the voltage dropped across the resistor increases. At 25 mA, the drop across the resistors is nearly 5-10% of the V_{SD} voltage for the 10 $k\Omega$ and 20 $k\Omega$ resistors, respectively. With a 5-10% decrease in the V_{SD}, the rate of injection is expected to decrease exponentially.

By performing a numerical differentiation on the data from Figure 14b, the rate of



(a)

(b)

(c)

Figure 14. This is a problem with the resistors

injection is determined, Figure 14c. Plotted against the drain current, the relationship between channel current and injection is clear; injection and the subsequent rate of injection are highly dependent on the channel current. The longer the device is injected, the more the channel current increases. In a situation where a high level of injection is desired, the channel current must constantly be reduced in order to facilitate continued injection at a desired rate. The most convenient method of modulating the channel current is accomplished by moving the gate input voltage. Accordingly, sweeping the gate coupling terminal of a floating-gate transistor, as discussed in Section 2.3.2.1, is a useful technique.

3.2.2.2 Rogue injection

Rogue injection is the name given to undesirable charge movement to un-selected devices in a floating-gate array during the programming process. The high voltages that result in

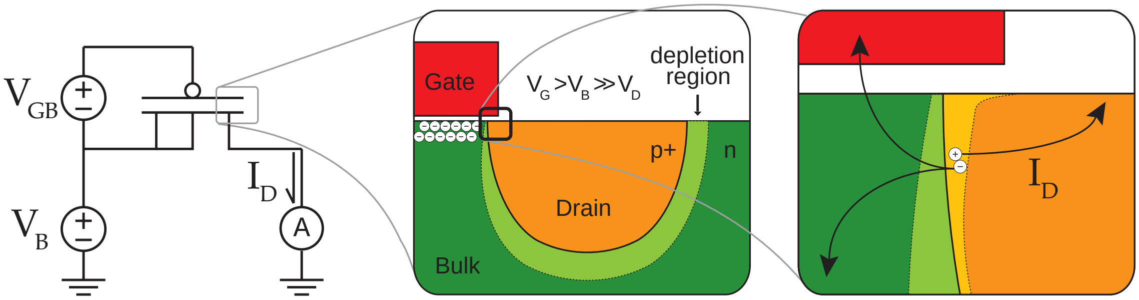


Figure 15. Conditions necessary to generate GIDL current in a pFET. The gate is brought above the bulk, the drain is held fixed at ground. Positive V_{GB} results in accumulation which pinches the depletion region around the drain. The high field and narrow depletion region allow electrons in the valence band of the drain to tunnel though the depletion region, illustrated by light-green and light-orange. When that occurs, holes are generated that move out to the drain. The electrons move toward the gate and bulk in a manner similar to DAHC injection.

DAHC injection can produce rogue injection when coupled with the isolation scheme illustrated in Figure 13. For un-selected devices on the same column as the selected floating-gate, the gated PN junction comprised of the active-to-gate overlap capacitor, the active itself, and the bulk beneath of the transistor's drain is the source of the problem. During the drain pulse, the un-selected transistor has a high gate voltage, a low drain voltage, and no channel. In addition, the transistor has an effective gate voltage above the bulk potential because of floating-gate charge. This causes accumulation to occur under the gate, as pictured in the center of Figure 15. As the drain voltage drops, the depletion width around the drain is narrowed near the accumulated surface under the gate. As the voltage difference between the drain and gate increases, the MOS capacitor can cause depletion or even inversion within the drain region under the gate. During this process, electrons in the valence band of the drain tunnel into the conduction band. The holes created are swept out the drain while the electrons head towards the bulk. The current generated is called gate-induced drain leakage [5], or simply GIDL. The electrons, which would normally be swept out the bulk, can inject into the conduction band of the SiO_2 because of the field created by the gate-to-drain voltage. In general, injection resulting from GIDL leads to majority carrier injection [31].

In order to measure the GIDL current, the schematic in the left portion of Figure 15

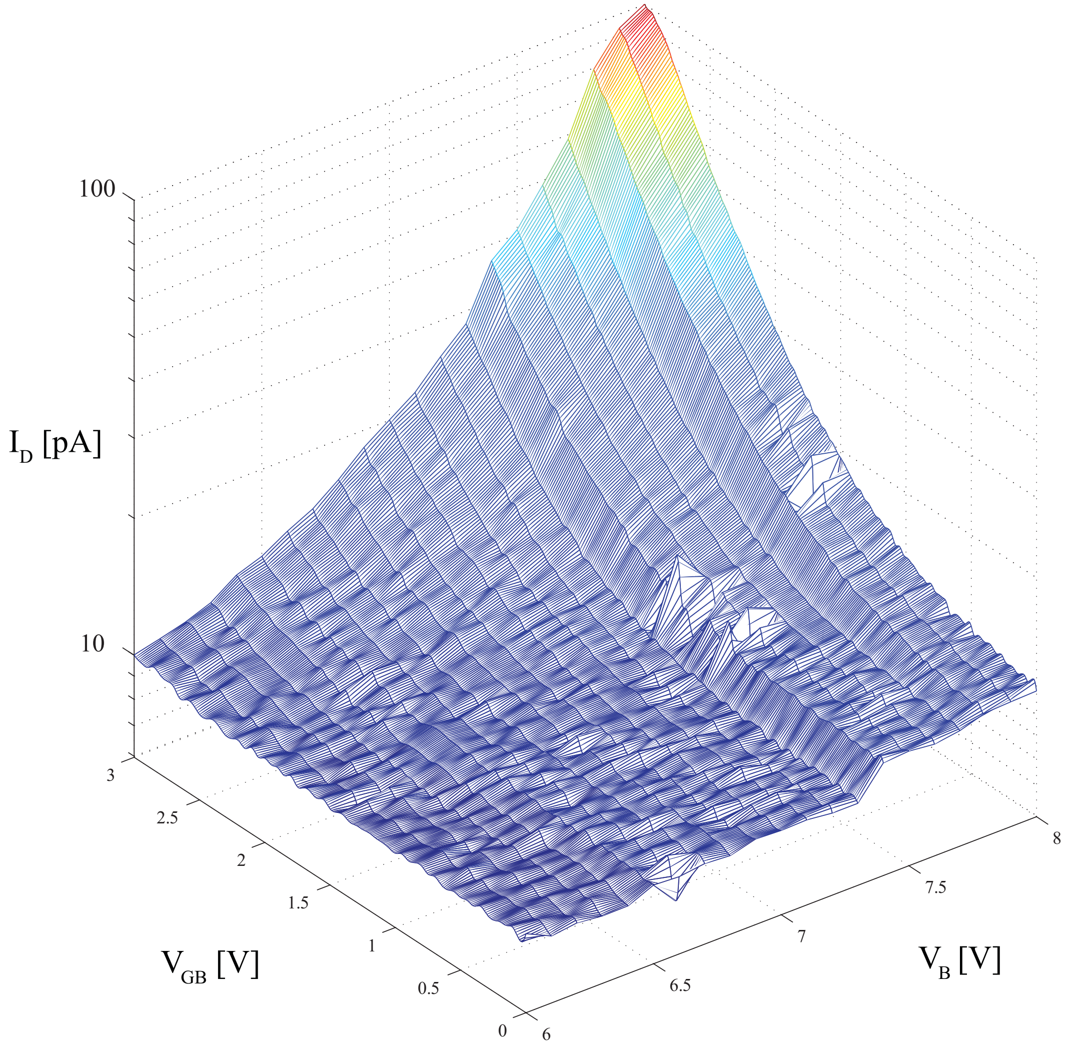


Figure 16. Experimental results for gate induced drain leakage. The bulk and source are tied together, the drain is connected to ground through an ammeter, and the gate is biased above the bulk potential. The resulting drain current for bulk voltages ranging from 6 to 8V and gate-to-bulk voltages from 0 to 3V is shown. The measurement floor of 10 pA is related to the reverse-bias current from the clamp-diodes protecting the drain terminal. Two useful conclusions are apparent from the experimental data. If there is less accumulation around the drain, the GIDL current is reduced. In addition, the bulk-to-drain voltage is also related to the leakage current measured—even with no accumulation, the current increases with bulk voltage.

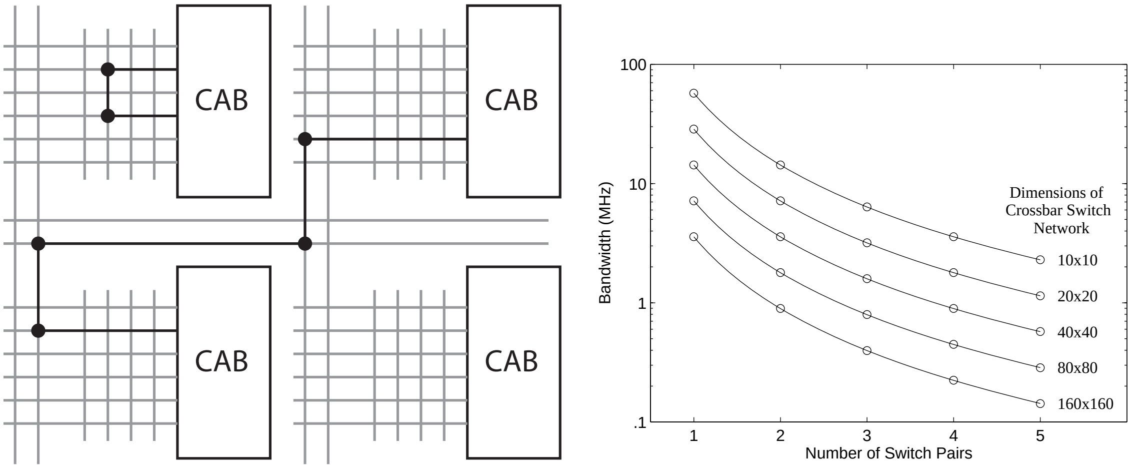
is used. The bulk and source are tied together, the drain is connected to ground through an ammeter, and the gate is biased above the bulk potential. The resulting drain current for bulk voltages ranging from 6 to 8V and gate-to-bulk voltages from 0 to 3V is shown in Figure 16. The measurement floor of 10 pA is related to the reverse-bias current from the clamp-diodes protecting the drain terminal. Two useful conclusions are apparent from the experimental data. If there is less accumulation around the drain, the GIDL current is reduced. In addition, the bulk-to-drain voltage is also related to the leakage current measured—even with no accumulation, the current increases with bulk voltage.

3.2.3 Bandwidth

A serious limitation to the bandwidth of an FPAA is the parasitic capacitance associated with its switching network. The analysis presented in this section has also been published in the context of characterizing a floating-gate switch network [6].

Based upon the architecture and layout of the crossbar network in [8], the capacitance contribution to a drain or source line per switch was calculated to be approximately 7 fF. When fully considering the capacitance contributions affecting a routed signal, all of the parasitic capacitances touching the signal line must be summed. In the case of the crossbar switch network, the capacitance contribution for each column is the number of rows multiplied by the parasitic capacitance of a single switch. Accordingly, the capacitance of a row is simply the contribution of a single switch multiplied by the number of columns, which is the number of switches along the row. Based upon this, the floating-gate pFET switch networks should have significantly less parasitic capacitance than a transmission gate based network and therefore be able to achieve higher bandwidths.

To better illustrate the effects of this parasitic capacitance, consider the FPAA architecture depicted in Figure 17a [8][9]. Local connections to and from the same CAB can generally be accomplished with a single pair of switches as seen in the upper left crossbar network in Figure 17a; CABs are discussed in Section 4. Routing between CABs is usually implemented through two pairs of switches unless the CABs are connected to the same set



(a) (b)

Figure 17. (a) Typical routing scheme used in FPAA crossbar switches. Each CAB has a local crossbar network for routing devices within the cab. These local networks connect to global routing lines that run vertically between the CABs. Horizontal global routing lines connect the vertical routing lines across the chip. (b) Switch network bandwidth for different routing lengths and crossbar configurations.

of vertical global routing lines, in which a single pair of switches will do.

Figure 17b shows the effects of the switch network dimensions and routing distance, defined as the number of switch pairs in the signal path, upon the bandwidth of the network. This plot uses a lumped model to estimate the effective bandwidth of the network given different routing scenarios. For expanded architectures with higher levels of routing beyond those in [8][9], routing distances greater than the expected 2 pairs of switches are examined. For the given 0.5 μm process, the expected bandwidth stays in the megahertz range for most reasonable network dimensions and routing lengths. However, Figure 17b clearly shows that FPAAs with higher routing levels must have carefully constructed networks to limit the number of switches per signal line in order to balance connectivity with bandwidth.

CHAPTER 4

COMPUTATIONAL ELEMENTS

Elements of computation represent the other half of a reconfigurable analog system; elements of connection are addressed in the previous chapter. Traditional elements of computation in analog circuits range from transistors and current mirrors to operational amplifiers and higher-level circuits. In a system augmented with floating-gate transistors, the same elements of computation tend to appear. However beyond the reconfigurability of circuit topology, floating-gates allow for reconfigurable circuit characteristics.

When implementing an FPAA, computational elements are typically clustered into groups in order to facilitate tradeoffs in routing flexibility and bandwidth as discussed in Section 3.2.3. The groupings are typically referred to as computational analog blocks, or CABs. This chapter is broken up into two sections relating to the two system platforms discussed in Section 5.

4.1 Analog circuits with floating-gate biases

As alluded to in the chapter introduction, a floating-gate analog circuit can simply be normal analog circuits with a floating-gate transistor providing a bias voltage or current.

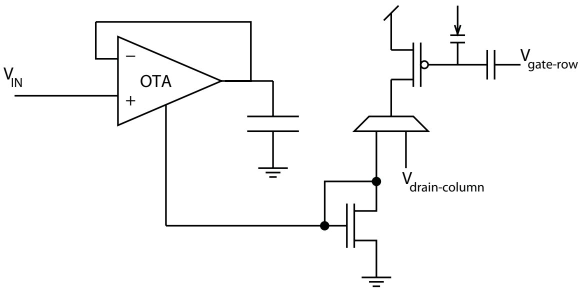
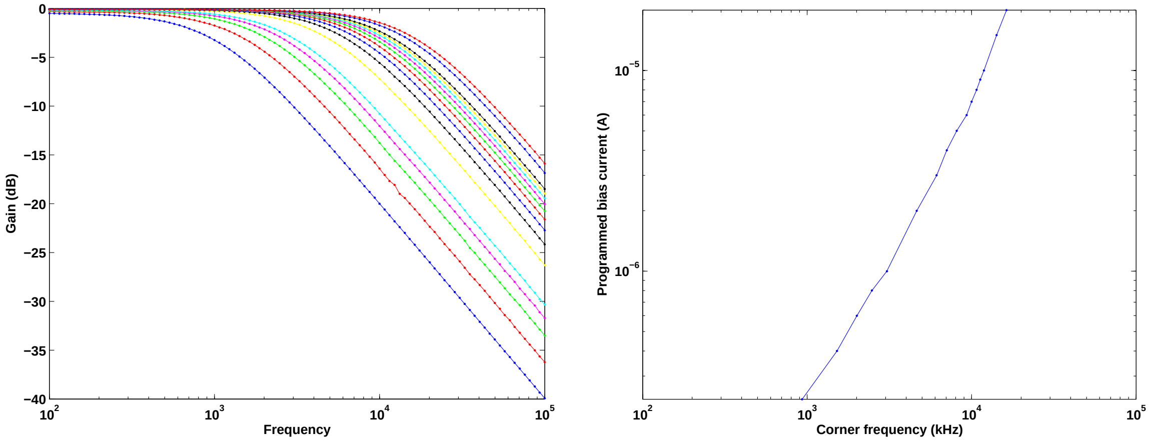


Figure 18. Floating-gate bias for a follower-integrator. The MUX on the drain of the bias transistor allows for switching between the programming circuitry of *prog* mode and computational circuitry of *run* mode.



(a) (b)

Figure 19. Follower-integrator frequency response and resulting bias map. Because g_m is proportional to bias current, and the bias current is set by a floating gate, the cutoff frequency of the amplifier can be set programmatically.

Floating-gate elements can be more integral to the computation; an entire class of adaptable circuits is described in [11] that use the charge stored on a floating-gate transistor as the adaptable element. In addition, the next section uses floating-gates as a more integral part of the computation. However, this section is only concerned with the framework of floating-gate transistors as peripheral elements.

The illustration in Figure 18 shows an operational transconductance amplifier connected as a follower with a tail current that is set by a floating-gate current source. The circuit, also referred to as a follower-integrator, integrates the OTA current on the capacitor. When an AC signal is provided at the input of the amplifier, an output signal will result that is attenuated at certain frequencies due to a pole given by $\frac{g_m}{C}$. Because g_m is proportional to bias current, and the bias current is set by a floating gate, the cutoff frequency of the amplifier can be set programmatically.

The MUX on the drain of the bias transistor allows for switching between the programming circuitry of *prog* mode and computational circuitry of *run* mode. When implementing large numbers of on-chip floating-gate biases, it is useful to take advantage of the array programming scheme of Section 3.2.1. Hence, the gate coupling line of the transistor is

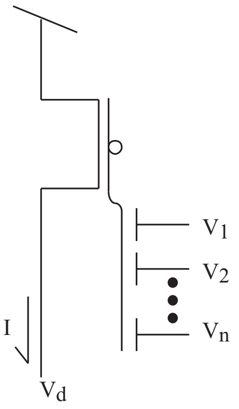


Figure 20. This is a symbol primarily used for a MITE in this document.

connected to a gate row, and the drain terminal is connected to a drain column.

Once a number of AC measurements have been taken for the follower-integrator, a mapping can be created between floating-gate bias currents and resulting cutoff frequencies. This is shown in Figure 19 [7]. The general framework for using floating-gate elements on the periphery to modify the characteristics of an analog circuit are the same. A figure of merit is related back to a measure of the charge stored on the floating gate, and the floating gate can then be programmed to a place where the desired characteristic is reached.

4.2 MITEs

A Multiple-input translinear element (MITE) is a circuit element that generates a current proportional to exponential of the linear weighted sum of its multiple voltage inputs. The MITE is useful in implementing circuits that adhere to the translinear principle [15, 18, 28]. MITE circuits are particularly well suited to reconfigurable analog arrays because they provide a regular, VLSI friendly architecture. In addition, there are a number of different methodologies which allow the direct mapping of algebraic expressions, ODEs, and other mathematical expressions to MITE networks [15, 28]. The symbol primarily used to represent MITEs in this document is shown in Figure 20.

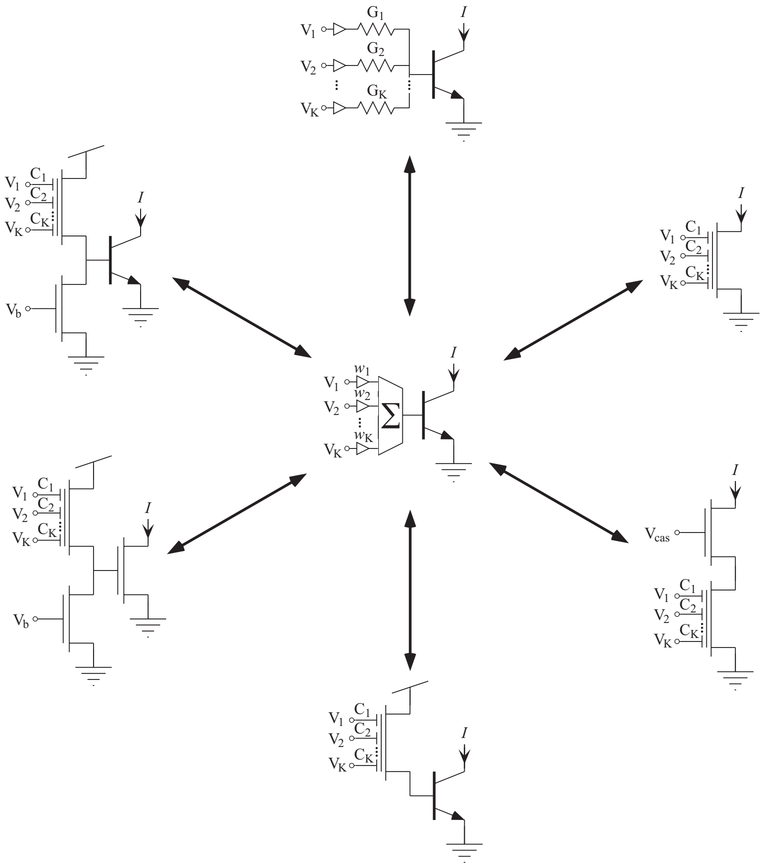


Figure 21. An ideal MITE and six different possible implementations.

4.2.1 Implementation

MITEs were conceived and developed by Brad Minch. In his doctoral thesis [15], he presents a number of different ways to implement MITEs. The ideal MITE is the center circuit of Figure 21. An ideal MITE has K input voltages. Each voltage is scaled by w_K, a positive dimensionless weight. The exponential result of the weighted sum is scaled and represented as a current. An idealized expression for the MITE is provided in Equation 8.

$$I = I_s e^{\sum w_i V_i} \tag{8}$$

Around the periphery of the ideal MITE in Figure 21 are six variations proposed in [15]. What follows is a discussion about two possible implementations.

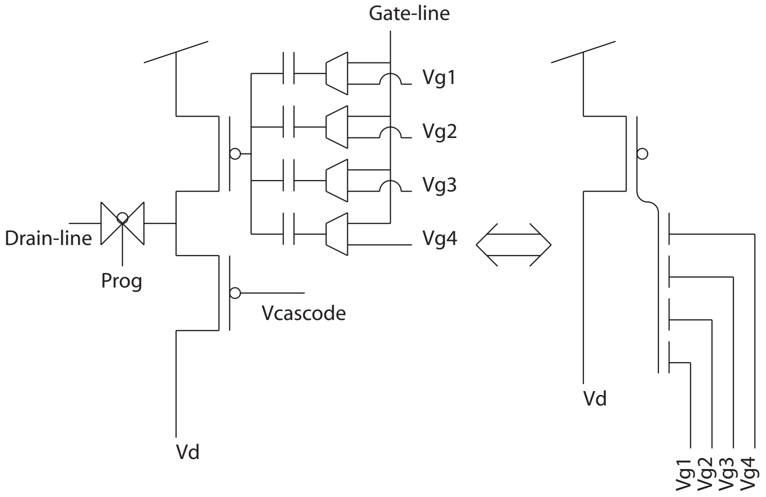


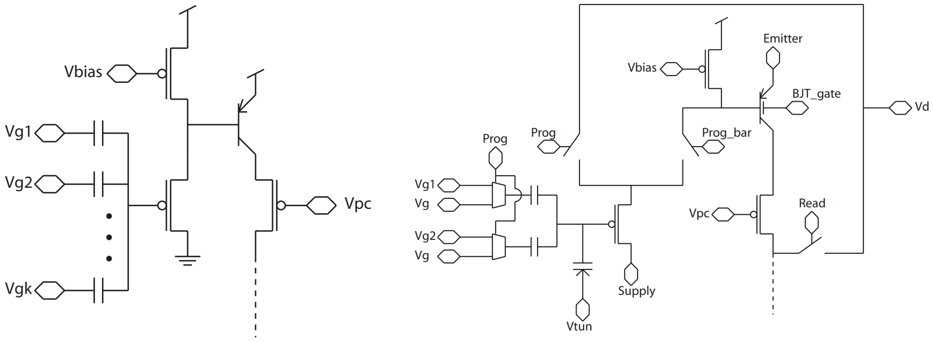
Figure 22. A sub-threshold 4-input MITE implementation and its mapping to the symbol in Figure 20.

4.2.1.1 *Sub-threshold floating-gate MITE*

The ideal MITE expression of Equation 8 is strikingly similar to Equation 1, the sub-threshold saturated drain current of a basic floating-gate transistor. As a result, a multiple-input floating-gate transistor can be almost directly implemented as a MITE. A four-input MITE is shown in Figure 22. A cascode is added to the floating-gate transistor to limit the drain-to-gate coupling. Assuming the dominant voltages coupling into the floating-gate node come from the explicit coupling capacitors, the expression for the MITE is provided in Equation 9.

$$I = I_s e^{\frac{\kappa V_{DD} - \kappa\left(\frac{C_1}{C_T}V_1 + \frac{C_2}{C_T}V_2 + \frac{C_3}{C_T}V_3 + \frac{C_4}{C_T}V_4\right)}{U_T}} \tag{9}$$

The MUXs on the V_G inputs of Figure 22 are necessary in order to fix the potentials to a single, known value during programming. The application of MITEs in an FPAA framework as presented here create a situation where the coupling voltages would otherwise be connected to drain lines of switch matrixes if MUXs were not used. The t-gate connected to the drain of the floating-gate provides a means for pulsing the drain to low potentials without the conductivity limitations of the pMOS cascode. In the context of a switch matrix, V_{G1-4} and V_d are connected to columns during *run* mode. The gate and drain lines are connected into the programming scheme similar to floating-gate bias transistor of Section



(a) (b)

Figure 23. BJT MITE concept and implementation.

4.2.1.

There are a number tradeoffs when using a subthreshold MITE with respect to the ideal form of Equation 8. First, the subthreshold range of a pMOS transistor is approximately two and a half decades. In addition, the lower end of the subthreshold current is in the pico to nano ampere range. As a result, systems of subthreshold MITEs must be very low noise and are limited in frequency. Furthermore, the κ term varies with V_G, making two MITEs with otherwise equal device characteristics have different κ values when different gate voltages are used.

4.2.1.2 BJT MITE

Another way to implement a MITE is to use a BJT as the exponential element, as alluded to in Figure 21. The BJT MITE circuit in Figure 23a also uses a floating-gate source-follower to implement a weighted sum of the input voltages. The major improvement of the MITE circuit in Figure 23a is that the exponential regime of the device is maintained over the entire operating range. As a result, the MITE can be operated at higher current levels.

The circuit can be analyzed as follows. Assuming the bias current of the follower has a negligible Early Effect and operates above threshold, KCL at the base of the BJT is given by

$$I_{bias} = \frac{I_0}{\beta} e^{\frac{V_{DD}-V_{base}}{U_T}} + \frac{K}{2\kappa}\left(\kappa\left(V_{DD} - V_{FG} - V_T\right) - \left(V_{DD} - V_{base}\right)\right)^2 \tag{10}$$

where V_{FG} is the standard weighted sum of input voltages scaled by $\frac{C_i}{C_T}$. Assuming the base current can be neglected, the base voltage is given by

$$V_{base} = \sqrt{\frac{2\kappa I_{bias}}{K}} + V_{DD} - \kappa\left(V_{DD} - V_{FG} - V_T\right) \tag{11}$$

Therefore, the output of the MITE is

$$I_{out} = I_0 \exp\left[\frac{\sqrt{\frac{2\kappa I_{bias}}{K}} + V_{DD} - \kappa\left(V_{DD} - V_{FG} - V_T\right)}{U_T}\right] \tag{12}$$

or more simply,

$$I_{out} = I_0' e^{\frac{\kappa V_{FG}}{U_T}} \tag{13}$$

This thesis focuses on CMOS compatible circuits, so a BJT-based MITE would need to be a lateral BJT. One of the limitations of a lateral BJT is the relatively low β. For small β, more power can be burned in the follower in order to maintain the assumption of Equation 11. In addition, the frequency response of the BJT MITE is damaged by low β since there is less current available. However, careful layout of the lateral BJT can improve the β and therefore the performance.

4.2.2 Synthesis proceedure

One of the powerful aspects of MITE circuits is that, like other translinear approaches, they map well to higher-level system descriptions. With respect to general translinear field, there is a lot existing work covering the synthesis of static and dynamic translinear circuits. A great deal of that work is summarized in [Dynamic Translinear Circuits - An overview]. In addition, at least two synthesis procedures have been developed specifically for MITEs [16, 28]. In [16], the synthesis procedure allows mapping from single output static polynomial constrains and algebraic differential equations to MITE circuits. Static and dynamic systems are treated in a similar manner. [28] relates multiple inputs and multiple outputs of a static mathematic expression to a connectivity matrix which is then mapped to MITEs. The dynamics of the system are mapped to first-order low-pass filters. It implements a smaller class of dynamic circuits, but provides a more direct mapping to an FPAA fabric.

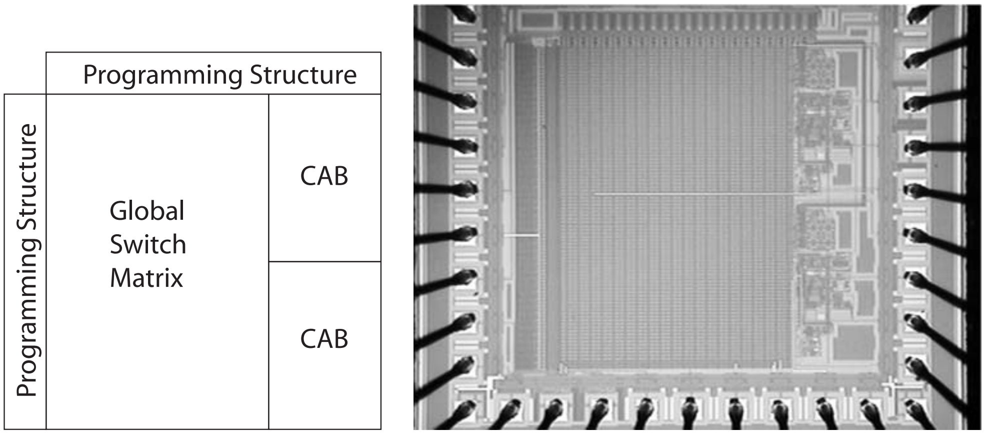
CHAPTER 5

COMPLETE SYSTEMS

The proceeding chapters have examined floating-gate transistors and their charge storage issues, switch networks, and circuits that benefit from floating-gate elements. This chapter takes the aforementioned concepts and brings them together into complete systems of field-programmable analog arrays. In order to analyze the quality of an FPAA, it is necessary to have a good sense of what an FPAA should do. The most fundamental goal of an FPAA is to provide a reconfigurable framework for implementing analog circuits and systems. Obviously, analog systems have specific performance metrics that allow for a straightforward evaluation. The evaluation of a circuit implemented in an FPAA is as straightforward as the investigation of a non-FPAA implementation, but the evaluation of reconfigurability provided by the FPAA is not so straightforward.

One way to discuss the reconfigurable quality of an FPAA is to talk about the number of computational analog blocks, or CABs. A CAB is the smallest unit of replication in an FPAA, and is made up of the computational primitives discussed in Section 4. Unfortunately, the size of a CAB is arbitrary, and so is the total computational complexity it represents.

While comparisons are difficult to make, the issues important to an FPAA design are straightforward. Beyond what has already been discussed, the level of granularity used in a CAB is a necessary consideration when building an FPAA. By making a computational primitive smaller, the number of potential uses the primitive represents increases. However, as the size of a primitive decreases, the number of switch lines necessary increases. Systems implemented over larger numbers of switches are exposed to greater switch parasitics, not to mention the size increase from a larger switch network. Therefore the performance of a system on an FPAA is roughly inversely proportional to the granularity of the computational primitive used to build a CAB.



(a) (b)

Figure 24. This is the architecture for the RASP 1.5. There are two CABs, a global cross-bar switch matrix, and programming circuitry. The contents of the CAB are displayed in Figure 25.

5.1 RASP

The Reconfigurable Analog Signal Processor, or RASP, is a platform for evaluating the design and implementation of floating-gate inspired FPAAs. Computational primitives of with varying levels of granularity are used in order to completely cover the design space. A focus is on providing maximum reconfigurability. The first RASP was comprised of two CABs and a full cross-bar switch [8]. The second implementation of the RASP was also comprised of two CABs and a full cross-bar switch, but had architectural improvements necessary to produce meaningful circuit and system data. In addition to the description that follows, the implementation and results of that chip have been published as [7].

Referred to as the RASP 1.5, it is the system illustrated in Figure 24. Each CAB has three amplifiers, three filter caps, a min and max detector, a bandpass, a pFET, an nFET, and a vector matrix multiplier, as shown in Figure 25. The OTA is a 9-transistor wide-range amplifier with a floating-gate bias current. The max and min detectors have floating-gate elements for varying the time-constant of the max and min detection decay. The bandpass is a cascade of two compact capacitively coupled current conveyors, or C^4's, with a buffer in between to reduce loading.

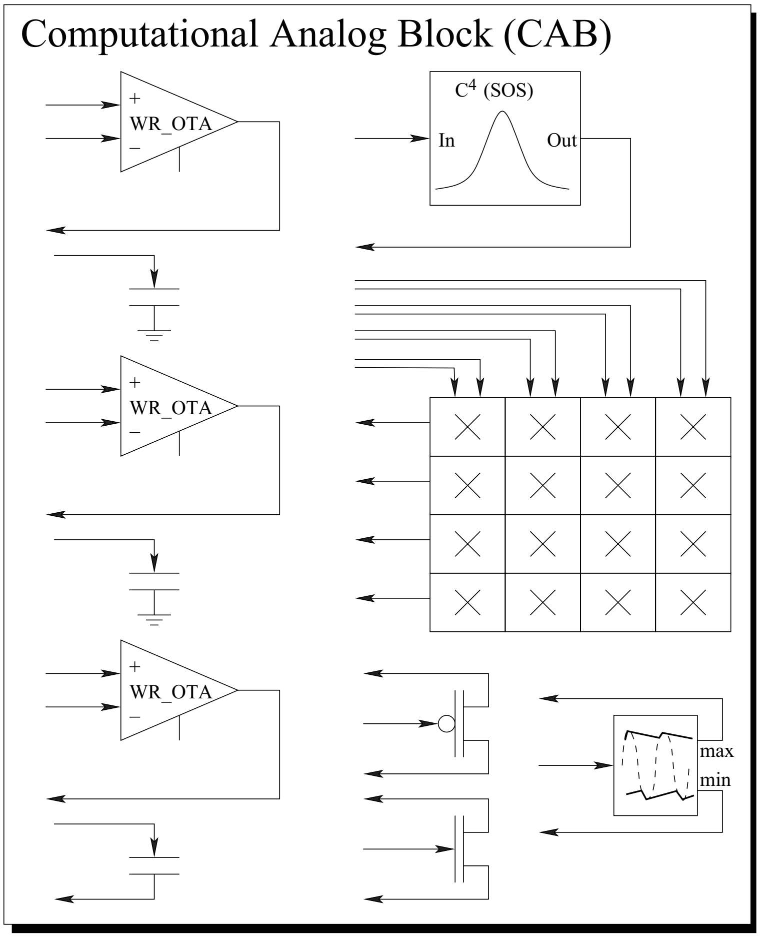


Figure 25. The components of a CAB on the RASP 1.5. The arrows represent connections to the switch matrix. The OTAs have a 9-transistor wide-range floating-gate biased architecture. The $C^4(SOS)$ block is a cascade of two C^4 circuits with a buffer in between. The max and min detectors have a τ set by a floating-gate.

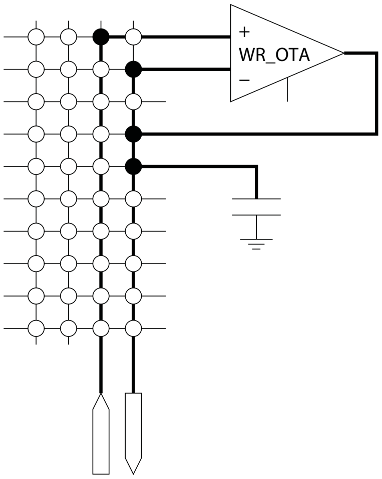


Figure 26. Implementation of a follower-integrator in a cross-bar switch matrix. The black dots represent switches that have been injected into the *on* position.

5.1.1 Follower-integrator

The implementation of follower-integrator is a good example of how to build circuits on a RASP FPAA. It is first necessary to tunnel the entire array in order to disconnect all of the switches and reset the bias positions in the array. Next, the circuit from Figure 18 is mapped to the switches in the connection matrix. The resulting mapping is shown in Figure 26. The programming algorithm used for switches, Section 2.3.2.1, is then used to make the necessary circuit connections. Finally, the bias current of the amplifier is set by programming the floating-gate transistor to achieve a particular corner frequency. In the case of the follower-integrator, the mapping from Figure 19b is used to facilitate the process.

5.1.2 Second-order Section

A slightly more complicated example of a circuit on the RASP is the second-order section, or SOS, pictured in Figure 27. This time all three OTAs and two caps from a single CAB are used to build the circuit. The circuit is desirable because it provides a low-pass transfer characteristic with a straightforward, predictable way to set the τ and Q factor. A small signal analysis of the circuit provides the means for relating the bias currents of the OTAs

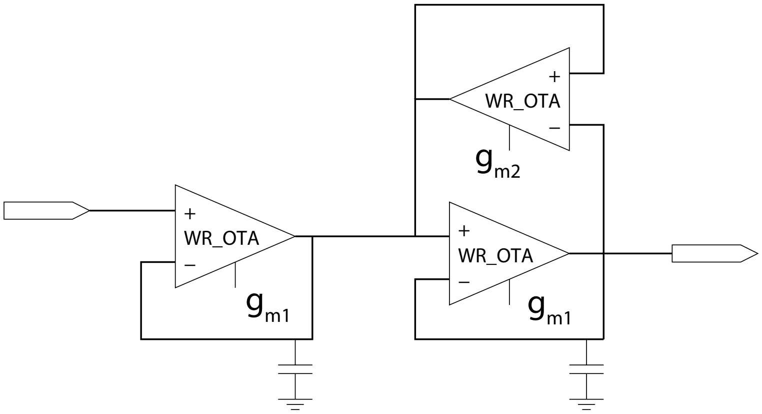


Figure 27. A second–order section filter can be implemented with two OTAs in a source-follower configuration and a third OTA that creates positive feedback.

to the transient response. The circuit has the transfer function

$$H(s) = \frac{1}{\tau^2 s^2 + \frac{1}{Q}\tau s + 1} \tag{14}$$

where

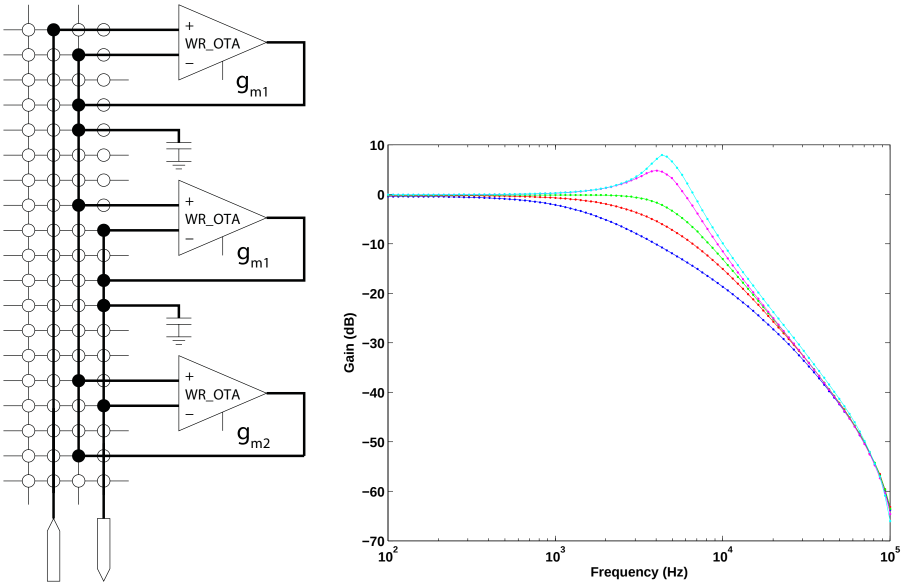
$$\tau = \frac{C}{g_{m1}} \tag{15}$$

and

$$Q = \frac{1}{2 - \frac{g_{m2}}{g_{m1}}} \tag{16}$$

The τ of the SOS is the same as the follower-integrator, so the mapping from Figure 19b is still valid as long as the same capacitance is maintained. If not, the τ would have to be determined experimentally. In order to set the Q factor, a ratio of transconductances is necessary, as demonstrated by Equation 16. In the case of two well matched, equivalently designed amplifiers, the ratio of transconductances is the ratio of the square-root of the bias currents. In subthreshold, it is simply the ratio of the bias currents.

The FPAA implementation and resulting data are shown in Figure 28. Data for a fixed g_{m1} and five different values of g_{m2} is shown. As expected, the Q factor increases with an increasing g_{m2}.



(a) (b)

Figure 28. SOS implementation and results. (a) The second-order section is implemented using the switch matrix, three OTAs, and two explicit capacitors. (b) The experimental frequency response of a circuit is shown here. Data for a fixed g_{m1} and five different values of g_{m2} is shown. As expected, the Q factor increases with an increasing g_{m2}.

5.1.3 Ladder filter

The availability of OTAs and grounded capacitors makes the RASP ideal for implementing Gm-C filters, as demonstrated in the previous section. One way to realize a particular filter is by modeling it with resistors, inductors, and capacitors, and then synthesize the design using Gm-C filters. In this example, a third-order Butterworth filter is implemented.

The canonical prototype of the filter, a double–resistance terminated LC filter, is shown in Figure 29a. By using the signal simulation method outlined in [29], the Gm-C filter shown in Figure 29b is generated. In order to maintain a maximally flat response, the following must hold: $2 * g_{m1} = g_{m2}$. Accordingly, the bias current of OTA-3 was set to half of the other OTA bias currents. A range of bias currents was used to create the frequency response shown in Figure 29c. As expected, the corner frequency of the filter is proportional to the bias currents of the OTAs. The lower corners were obtained by using a bias current in the range of hundreds of pico-amps, while the highest corners required currents of up to 1 μA.

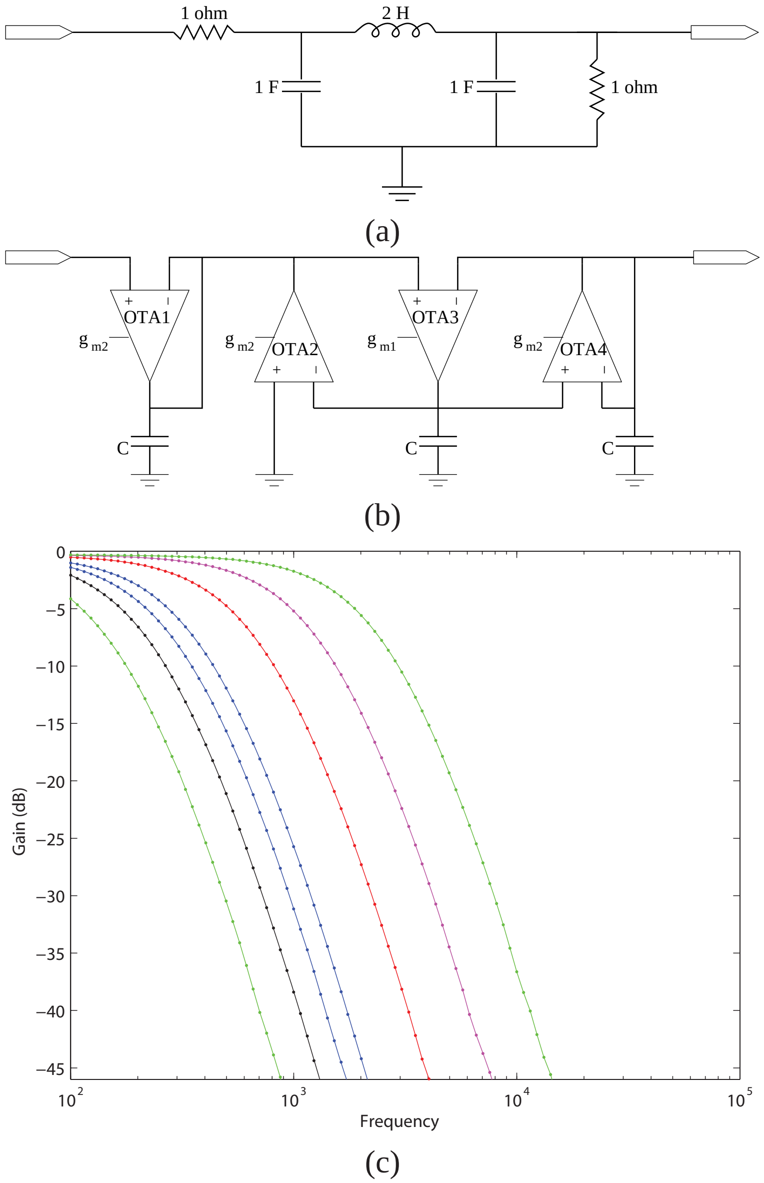
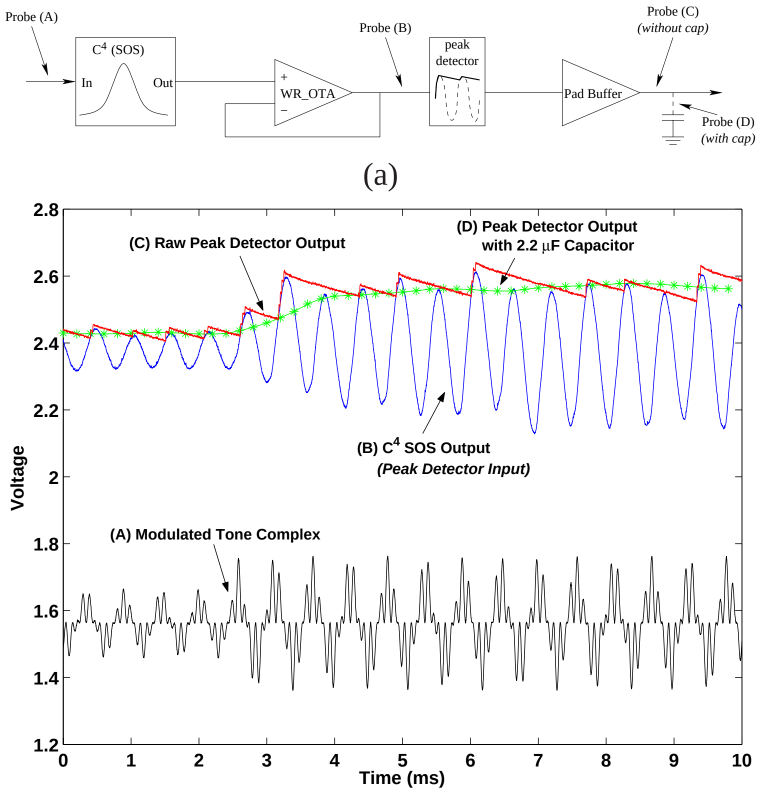


Figure 29. (a) The canonical prototype of a third–order Butterworth double–resistance terminated LC filter. (b) This is the Gm-C implementation of the same filter. The filter can be realized directly on the RASP 1.5 FPAA. (c) Results from the ladder filter for diffferent bias currents.



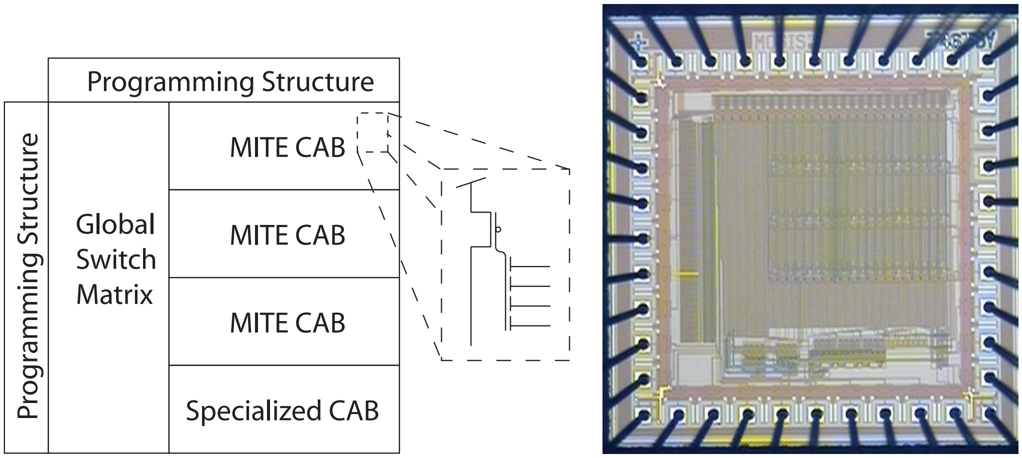
(a)

(b)

Figure 30. A C^4 SOS block, an OTA, and a peak detector are connected in series in order to calculate the energy of a signal within bandpass of the SOS.

5.1.4 Subbanding

This next section is an example of a larger system element composed of coarse-grained CAB components. Specifically, the C^4 SOS block, an OTA, and a peak detector are connected in series as pictured in Figure 30a. The C^4 module selects a particular band, the OTA prevents loading, and the peak detector calculates the envelope of the signal. As shown in Figure 30b, the input is an amplitude-modulated signal with 1.8 KHz and 10.0 KHz frequency components. The C^4 module is biased to have a center frequency near 1.8 KHz, and the OTA is configured to be a noninverting buffer. The output of the system is shown in Fig. 30b. Also, the output of the system is shown after an external 2.2 μF capacitor has been added at the output of the FPAA. This change has the effect of smoothing (i.e., low-pass filtering) the output, thus creating a longer effective time constant for the system.



(a) (b)

Figure 31. System architecture of the RAAM 1, an FPAA used to create reconfigurable translinear networks. The system consists of 3 MITE CABS, a specialized CAB, and a global switch network. The specialized cab consists of circuitry that enables dynamic functions and also includes an input bank of V-I converters.

The effect of the aforementioned circuit is to calculate the energy of a signal within the chosen frequency band. When many bands are chosen, the result approaches the magnitudes of a discrete fourier over time.

5.2 RAAM

The Reconfigurable Analog Array of MITEs, or RAAM, is another test-bed FPAA built upon the same switch technology as the RASP. However unlike the RASP, the RAAM represents a concerted effort to better mimic FPGA design by using a regular computational primitive, explicitly supporting direct system synthesis, and having multiple levels of connection hierarchy. As discussed previously, MITEs lend themselves well to VLSI implementation and leverage the power of translinear circuit methodology. The focus of this chapter is the RAAM 1, an initial MITE FPAA implementation which has been published about as [1]. The RAAM is provided a version number as a means for distinguishing it from the large-scale MITE FPAA discussed in Section 6.2.

The RAAM 1 has four CABs, a global switch matrix, and programming circuitry. It is pictured in Figure 31 and was fabricated in a .5μ process. The global switch matrix is

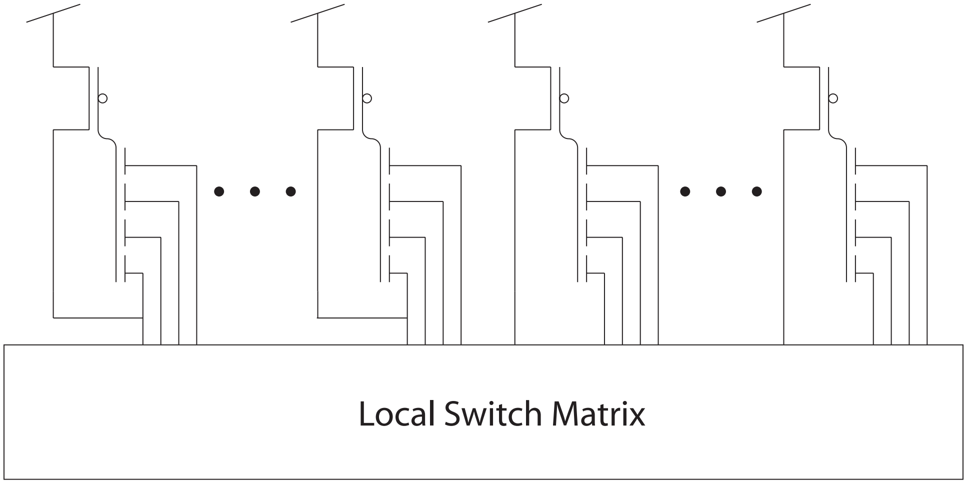


Figure 32. RAAM CAB architecture. The CAB has a local switch matrix which makes up the first level of routing. There are two computational primitives in the CAB, an input MITE with a diode connection and an output MITE.

actually the second layer of hierarchy–the MITE CAB pictured in Figure 32 is composed of MITE primitives and a local switch matrix. The purpose of the local network is provide a trade-off between switch area and reconfigurability. There are two analog primitives in the MITE CAB, a diode connected MITE for input signals and an plain MITE for output signals. The 4-cap structure was chosen as a means for mapping cleanly to the synthesis procedure in [28]. The specialized CAB contains the I-V converters for inputs and first-order low-pass filters for implementing dynamic circuits.

The following experimental results illustrate the basic functionality of the RAAM architecture.

5.2.1 Single-input power-law circuit

One of the qualities a MITE is fundamentally good at is implementing power-law equations. Accordingly, a good starting point with the RAAM is the implementation of a circuit that results in an input raised to a particular power. Using [28], $y = x^2$ can be represented as

$$I_{out} = \frac{I_{in}^2}{I_{ref}} \tag{17}$$

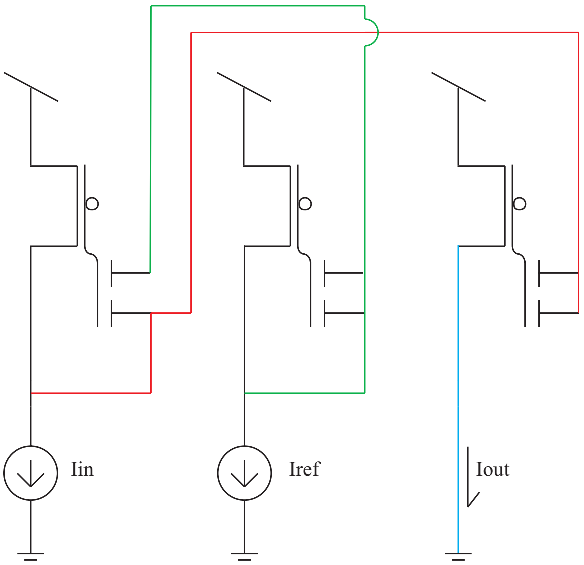


Figure 33. Schematic of a squaring circuit represented by Equation 17 using two input MITEs and a single output MITE. The coloration corresponds to connections made in order to implement the circuit on the RAAM 1. The mapping is illustrated in Figure 35

where I_{out} is the output current, I_{in} is an input current, and I_{ref} is a scaling current which represents unity. The circuit that represents Equation 17 is shown in Figure 33. In order to verify the operation of the circuit, Equation 9 is first reduced to two gate capacitors and is rearranged into

$$I_{out} = I_s' e^{wV_1 + wV_2} \tag{18}$$

where

$$I_s' = I_s e^{\frac{\kappa V_{DD}}{U_T} \frac{Q}{C_T}} \tag{19}$$

and

$$w = -\frac{\kappa}{U_T}\left(\frac{C}{C_T}\right) \tag{20}$$

By defining V_{ref} as the diode-connected voltage created by I_{ref} and V_{in} as the diode-connected voltage created by I_{in}, the following expressions for the controlling voltages

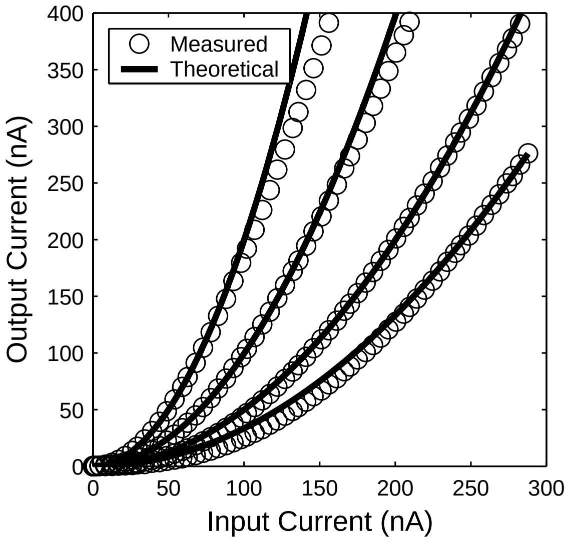


Figure 34. Experimental results of a MITE squaring circuit plotted against simulation data. Reference currents of $50nA$**,** $100nA$**,** $200nA$**, and** $300nA$ **were used.**

can be written from the circuit in Figure 33 by applying Equation 18.

$$V_{ref} = \frac{1}{2w}\ln\left(\frac{I_{ref}}{I'_s}\right) \tag{21}$$

$$V_{in} = \frac{1}{w}\ln\left(\frac{I_{in}}{I'_s}\right) - V_{ref} \tag{22}$$

As a result,

$$\begin{aligned} I_{out} &= I'_s e^{2wV_{in}} \\ &= I'_s e^{2\ln\left(\frac{I_{in}}{I'_s}\right) - \ln\left(\frac{I_{ref}}{I'_s}\right)} \\ &= I'_s \left(\frac{I_{in}}{I'_s}\right)^2 \left(\frac{I'_s}{I_{ref}}\right) \end{aligned} \tag{23}$$

which can be simplified to Equation 17.

The analysis yields important insight. In order to cleanly simplify the result, the weight term must be equal for all three MITEs. If there is any variation in the weight term, it appears in the final expression as an additional exponent in the same way the squared term does. Accordingly, it is necessary to maintain good capacitor matching and avoid large temperature gradients across the circuit operation. In addition, offsets in the charge show up as a gain term, as discussed in Section 5.2.3.

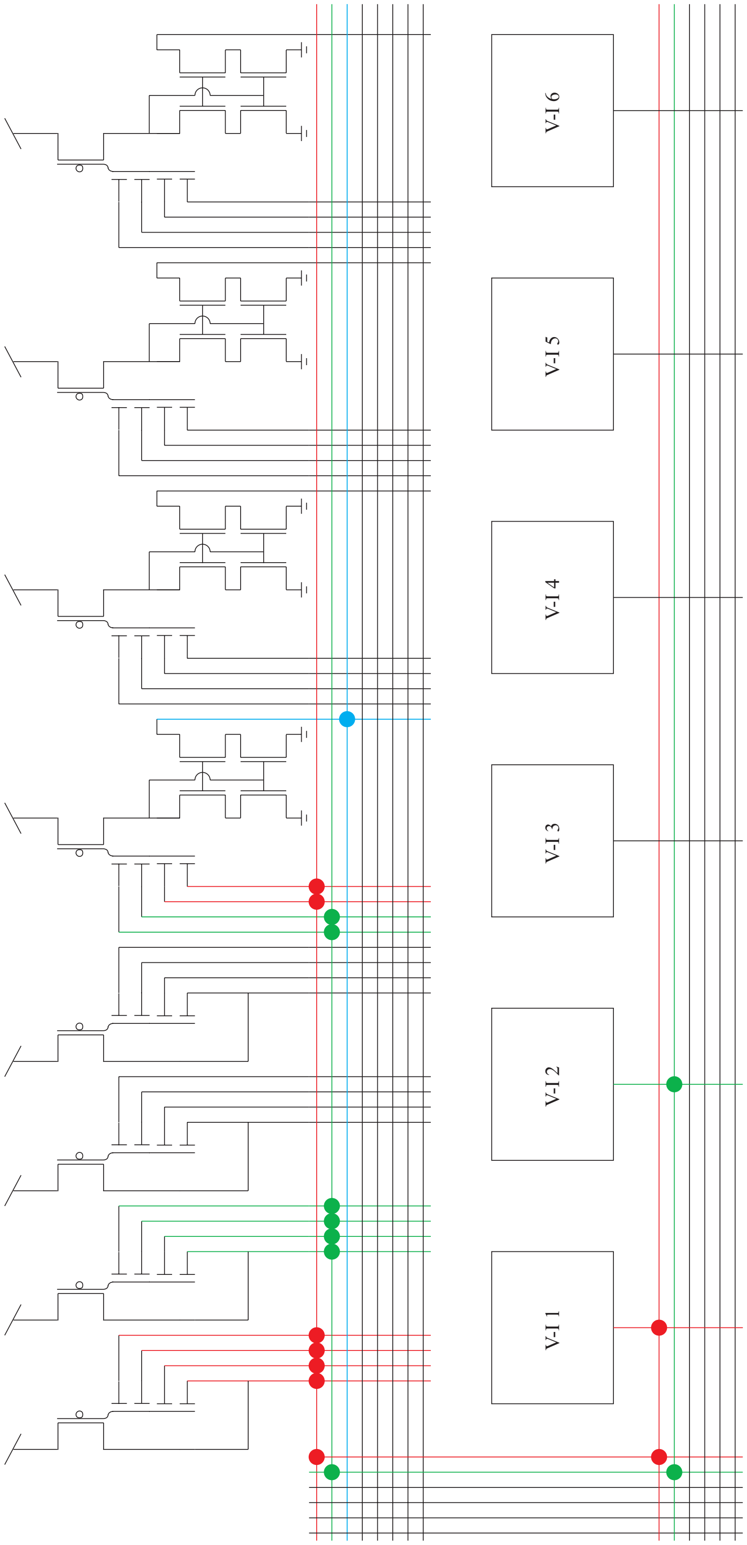
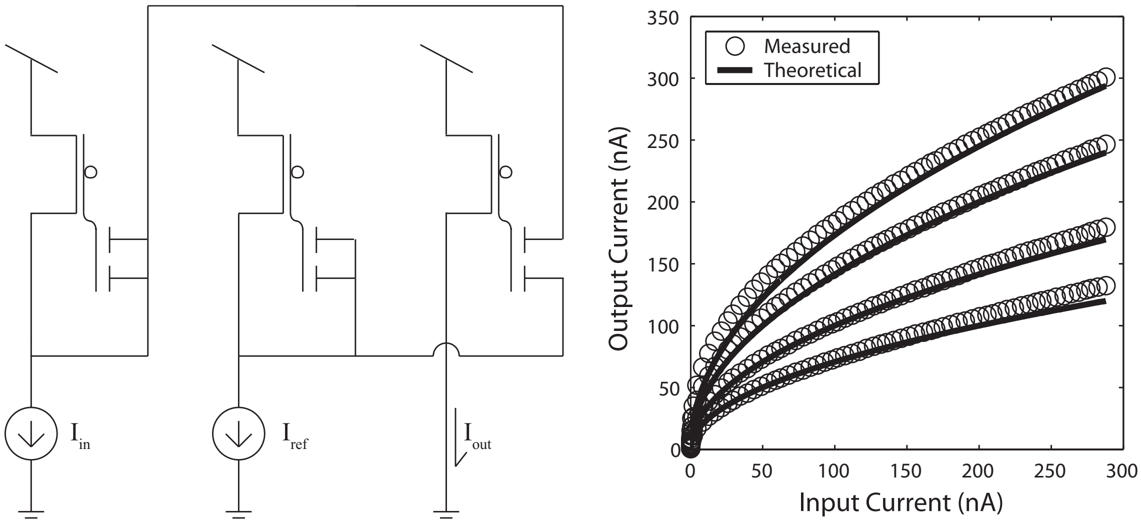


Figure 35. Example of RAAM 1 reconfigured to implement a squaring circuit. The colored nodes correspond to Figure 33 and the circles at the intersection of the bus lines indicates a switch that has been turned on. The row of V-I converters and the crossbar network below it represent the specialized CAB, the crossbar network on the left of the figure represents the global switch matrix, and the row of MITEs and the crossbar network below it represent a MITE CAB.



(a) (b)

Figure 36. Circuit and experimental results of a MITE square-root circuit. Reference currents of 50*nA***,** 100*nA***,** 200*nA***, and** 300*nA* **were used.**

The square circuit was compiled into the RAAM yielding the experimental data plotted against simulation data in Figure 34. Reference currents of 50*nA*, 100*nA*, 200*nA*, and 300*nA* were used. The circuit is implemented by mapping the 2-cap MITE circuit to 4-cap MITEs in a RAAM CAB. The resulting implementation is shown in Figure 35. Two currents are routed to two input MITEs and an output MITE. The colored circles at line intersections represent switches that have been injected to the on position. The output MITE uses a cascoded nFET current mirror in order to reduce distortion in the current mirror. Variations due to differences in the current mirror can be taken care of by varying the charge on the output MITE.

By varying the switch connections, the same set of MITEs can implement a square-root. The MITE circuit and resulting experimental data are shown in Figure 36, where

$$I_{out} = \sqrt{I_{in} I_{ref}} \tag{24}$$

is implemented. The square-root circuit tracks the theoretical data better than the squaring circuit for the same set of reference currents because of the subthreshold range of the MITEs. An expansive operation like a power term greater than one will necessarily leave the valid operating region before a compressive one.

5.2.2 Translinear loop

A translinear loop is a balanced equality of a product of currents where the currents are related to the exponential of a controlling voltage. A translinear loop is a powerful circuit tool for implementing multiplications, divisions, and power-law equations. The functional restraint for static equations a loop can implement is that the expression must take the form

$$\prod_{i=1}^{n} I_i = \prod_{j=1}^{m} I_j \tag{25}$$

where n and m are equal. An example 2^{nd}-order translinear loop is

$$I_1 I_3 = I_2 I_4 \tag{26}$$

The circuit implementation of the loop is shown in Figure 37 and the output current I_4 is given as

$$I_4 = \frac{I_1 I_3}{I_2} \tag{27}$$

By using the same analysis as in Section 5.2.1, the controlling voltages can be determined.

$$V_1 = \frac{1}{w} \ln\left(\frac{I_1}{I_s'}\right) - V_{ref} \tag{28}$$

$$V_2 = \frac{1}{w} \ln\left(\frac{I_2}{I_s'}\right) - V_1 \tag{29}$$

$$V_3 = \frac{1}{w} \ln\left(\frac{I_3}{I_s'}\right) - V_2 \tag{30}$$

By plugging Equations 28–30 into 18 for I_4,

$$\begin{aligned} I_4 &= I_s' \exp\left\{wV_{ref} + w\left[\frac{1}{w}\ln\left(\frac{I_3}{I_s'}\right) - \frac{1}{w}\ln\left(\frac{I_2}{I_s'}\right) + \frac{1}{w}\ln\left(\frac{I_1}{I_s'}\right)\right] - V_{ref}\right\} \\ I_4 &= I_s' \exp\left\{\ln\left(\frac{I_3}{I_s'}\right) - \ln\left(\frac{I_2}{I_s'}\right) + \ln\left(\frac{I_1}{I_s'}\right)\right\} \end{aligned} \tag{31}$$

which simplifies down to Equation 27. The loop circuit was implemented on the RAAM in the form expressed in Figure 38. In order to build the current without an explicit voltage

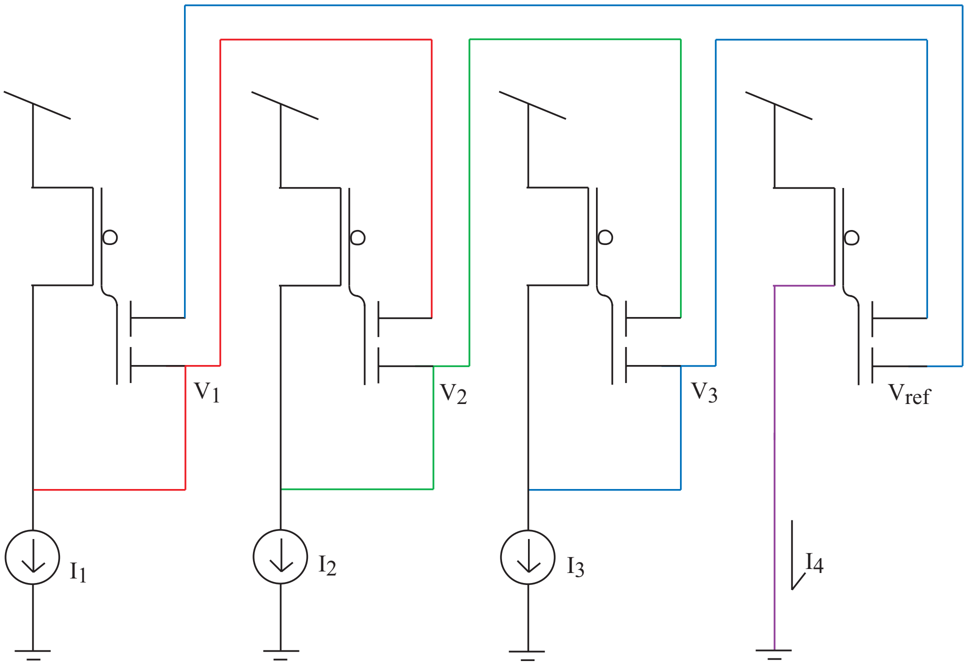


Figure 37. MITE implementation of a 2^{nd}-order translinear loop. The coloration corresponds to connections made in order to implement the circuit on the RAAM 1. The mapping is illustrated in Figure 38.

source for V_{ref}, the controlling voltage for the output current was used. A series of sweeps for various multiplicative coefficients was performed on the loop circuit. The resulting data is plotted in Figure 39. As the input leaves the deep subthreshold region of the MITE, the output begins to bend.

5.2.3 Vector magnitude

As an example of a slightly more complicated system, a MITE circuit that calculates the vector magnitude was compiled onto the RAAM. The equation for the circuit is given by

$$I_{out} = \sqrt{I_x^2 + I_y^2} \tag{32}$$

The inputs provided to the system in the form

$$I_x = I_{ref} * cos(\theta) \tag{33}$$

$$I_y = I_{ref} * sin(\theta) \tag{34}$$

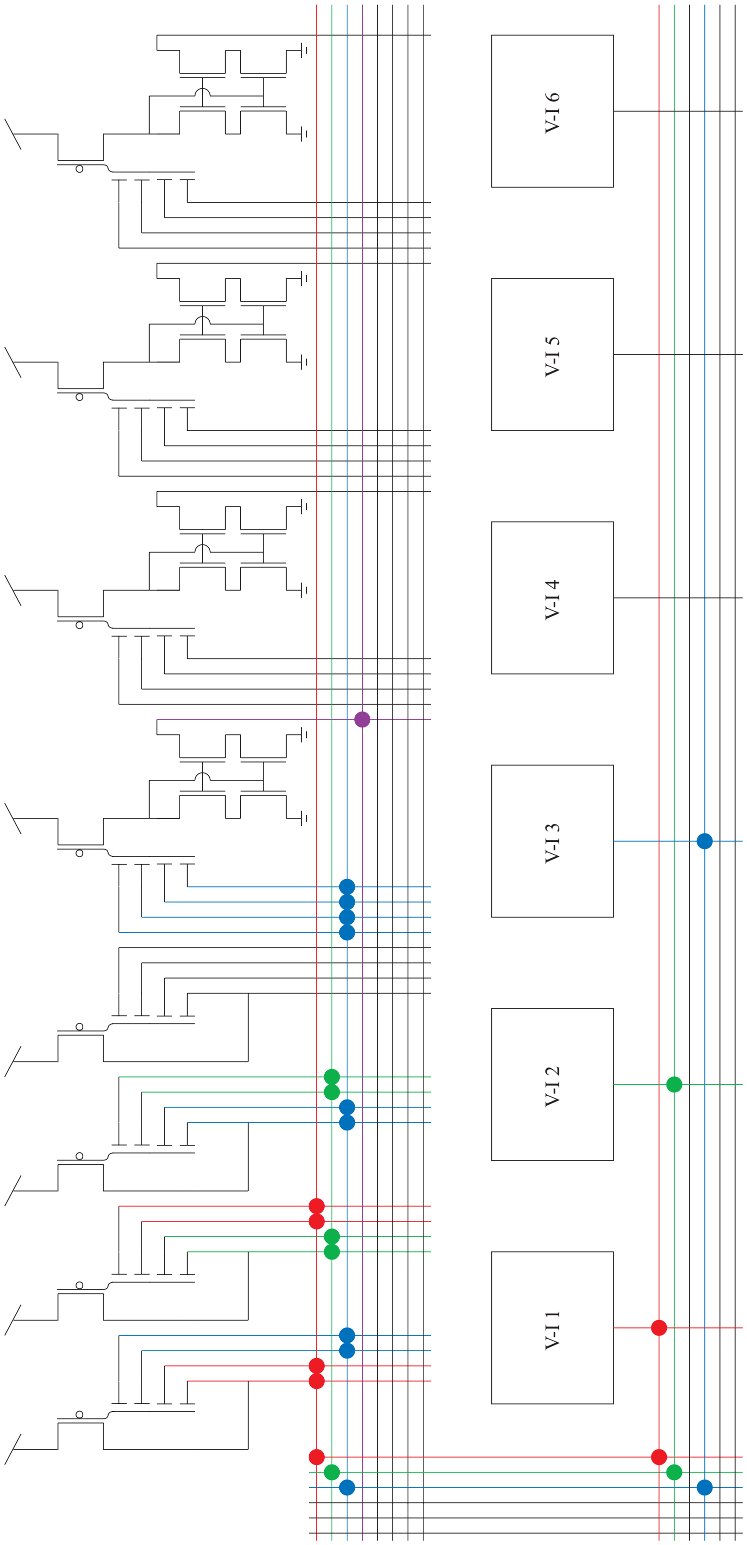


Figure 38. Example of the RAAM1 reconfigured to implement a $2nd$**-order translinear loop. The circuit schematic is shown in Figure 37. The circles at the intersection of the bus lines indicate a switch that has been turned on.**

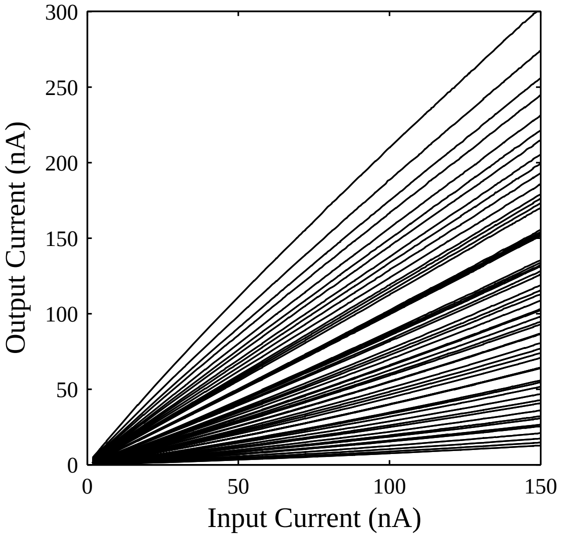


Figure 39. Results of a 2^{nd} order translinear loop for various multiplicative coefficients

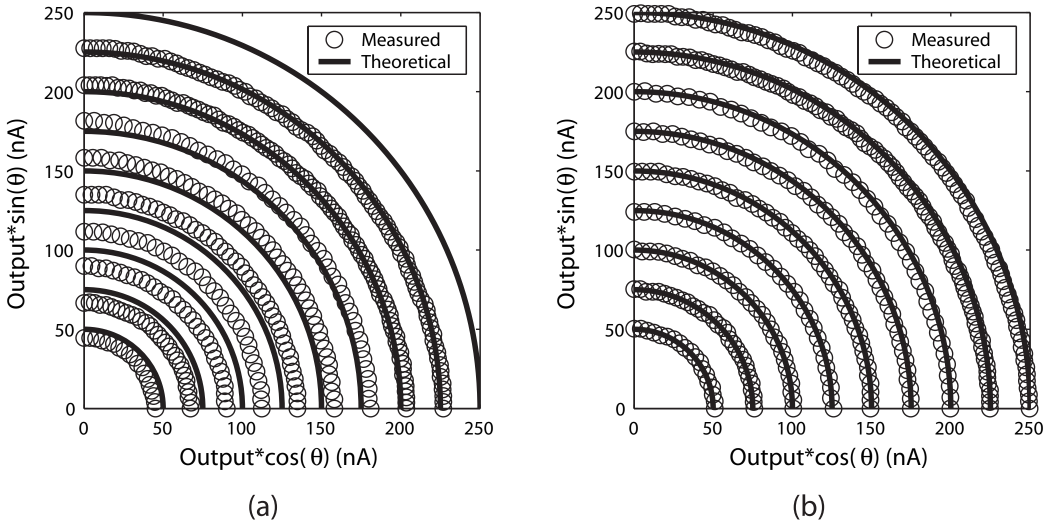
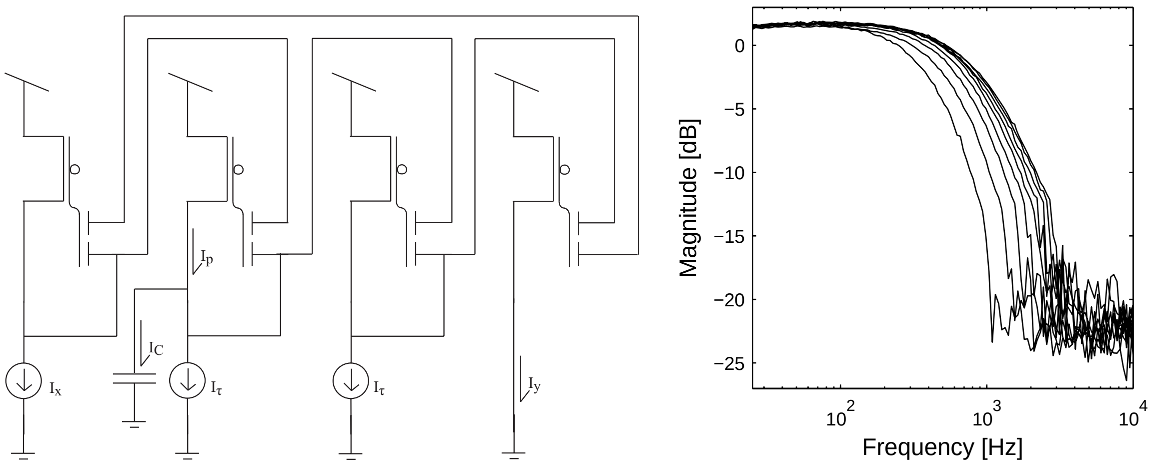


Figure 40. Results of the vector magnitude circuit. (a) Results of the vector magnitude circuit after programming all MITEs to the same level. Each MITE was programmed to have $10nA$ of current with a source-drain voltage of $2.3V$ and a source-gate voltage of $1.3V$. (b) Results of the vector magnitude circuit after programming out the initial errors. The MITEs preforming the squaring functions were injected higher than the other MITEs in order to increase the coefficients to 1.



(a) (b)

Figure 41. MITE implementation and experimental data of a 1^{st}-order low-pass filter. The filter is simply a translinear loop with a capacitor on an internal node.

where θ is swept from 0 to 90°. A plot of the initial system implementation is shown in Figure 40a. Each concentric arch represents a different I_{ref}. Empirically, the data has a coefficient under the square-root that is modeled as

$$I_{out} = \sqrt{0.8I_x^2 + 0.8I_y^2}. \tag{35}$$

By increasing the charge of the MITEs implementing the squaring terms, a gain can be applied to achieve an effect of unity. The resulting date is shown in Figure 40b.

5.2.4 First-order filter

MITEs can be used to implement more than static equations. By adding a 1^{st}-order low-pass filter to a static MITE network, algebraic differential equations can be implemented [16, 28]. If I_x is an input current and I_y is an output current equation, a low-pass filter can be expressed as

$$\tau\frac{dI_y}{dt} + I_y = I_x \tag{36}$$

However, capacitors are related to current through the time-derivative of a voltage, so the chain rule is used to rearrange (36) into

$$\tau\frac{\delta I_y}{\delta V_y}\frac{dV_y}{dt} + I_y = I_x \tag{37}$$

In addition, the voltage-derivative of I_y is simply the g_m of a MITE, resulting in

$$-w\frac{\tau I_y}{U_T}\frac{dV_y}{dt} + I_y = I_x \tag{38}$$

After introducing $\frac{C}{C}$ to the time-derivative of V_y and rearranging,

$$I_\tau - I_C = \frac{I_x I_\tau}{I_y} = I_p \tag{39}$$

Accordingly, (39) can be directly implemented as a translinear loop with a capacitor on one of the inputs.

The resulting circuit implementation is shown in Figure 41a. The filter was implemented on the RAAM 1 with the resulting experimental data shown in Figure 41b.

CHAPTER 6

CONCLUSION

In this document I have explained how floating-gate transistors work, how they allow us to approach reconfigurable circuits, and how we can use them to construct field programmable analog arrays. I demonstrated that the floating-gate is competitive with a t-gate, and that if switch transition time is not an issue, the floating-gate switch is superior. I have presented two alternative views of an FPAA, the RASP and the RAAM. The RASP is useful for analog designers who want a cache of classical analog components for circuit building. The RAAM is more useful for tying to a synthesis environment. As we become more savvy at instrumenting our floating-gate systems, I would imagine the RAAM bridging the gap between theoretical system work and analog system implementation.

6.1 Personal Contributions

I contributed to the design and implementation of the programming circuitry on the RASP 1.5. I designed and implemented the OTA used in the CABS. I then took that programming architecture from the RASP and used it to construct the RAAM. I contributed to every aspect of the RAAM: the design, implementation, and experimental verification. I also changed the way we interact with floating-gate switches by recognizing that switch isolation is not necessary for switch programming. I directly contributed to all of the data taken off of the RAAM. On the RASP however, the only data I was personally responsible for was the ladder-filter data. The GIDL and drain parasitic data was mine exclusively. I contributed to the floating-gate retention data. I also directly contributed to the construction of the BJT MITE presented.

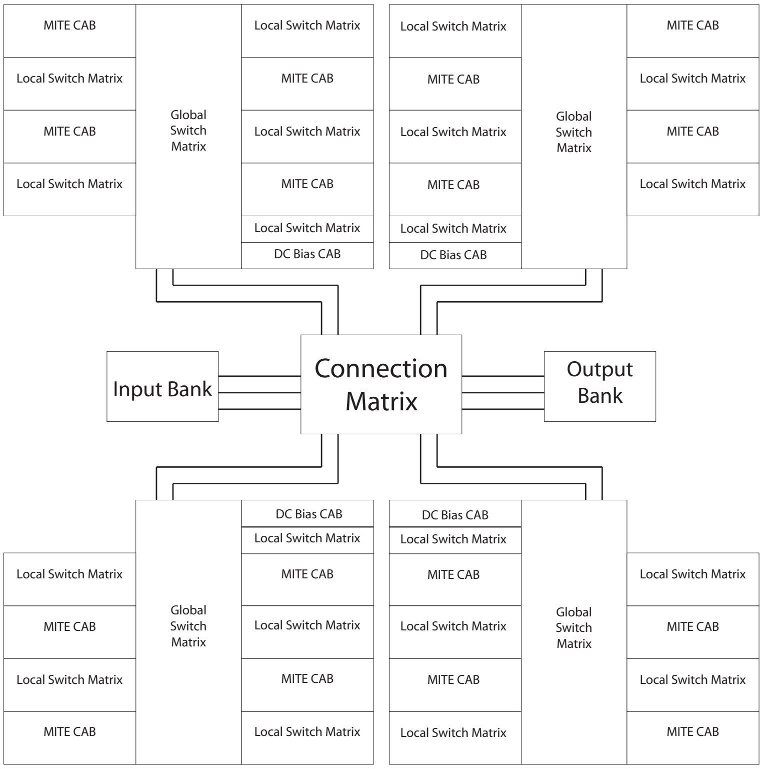


Figure 42. Architecture for the RAAM 2.

6.2 Future Work

Beyond the work presented herein, I have continued to work on building large and more complex FPAAs. I have worked on and designed a large-scale MITE FPAA referred to as the RAAM 2. Shown in Figure 42, the RAAM 2 is the architectural equivalent of four RAAM 1's, which I will refer to as sectors. Each sector has its own switch matrix and CABs. The sectors are linked by a common cross-bar switch matrix. The connection matrix is also the I/O hub of the chip. In order to reach higher frequencies and increase the current range of the MITE, the RAAM 2 uses BJT MITEs exclusively.

The CAB structure has also changed in the RAAM 2. Shown in Figure 43, the new cab structure is made up of loops, adders, and subtractors. Specifically, there is a 3^{rd}-order loop, two 2^{rd}-order loops, a 2^{rd}-order loop with a capacitor, three adders, and two subtractors. The loop with a capacitor is shared with one of the 2^{rd}-order loops in order to reduce switch column.

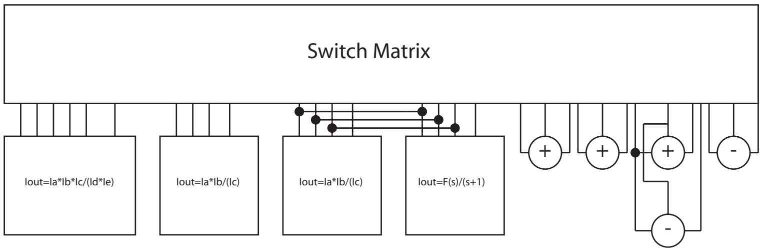


Figure 43. RAAM 2 CAB structure. The inputs for the loop with a capacitor are shared with one of the 2^{rd}-order loops in order to reduce the number of switch columns. The same tradeoff is made with the an adder-subtractor pair.

REFERENCES

[1] Abramson, D., Gray, J., Subramanian, S., and Hasler, P., “A field-programmable analog array using translinear elements,” *IEEE International Workshop on System-on-Chip for Real-Time Applications (IWSOC)*, 2005.

[2] Becker, J. and Manoli, Y., “A continuous–time field programmable analog array (FPAA) consisting of digitally reconfigurable G_M–cells,” in *Proceedings of the International Symposium on Circuits and Systems*, pp. 1092–1095, 2004.

[3] Benson, R. and Kerns, D., “Uv-activated conductances allow for multiple time scale learning,” *IEEE Transactions on Neural Networks*, vol. 4, pp. 434 – 40, May 1993.

[4] Bleiker, C. and Melchior, H., “A four-state EEPROM using floating-gate memory cell,” *IEEE Journal of Solid-State Circuits*, vol. 22, pp. 460–463, June 1987.

[5] Chen, J.-H., Wong, S.-C., and Wang, Y.-H., “An analytic three-terminal band-to-band tunneling model on gidl in mosfet,” *IEEE Transactions on Electron Devices*, vol. 48, pp. 1400 – 5, July 2001.

[6] Gray, J., Twigg, C., Abramson, D., and Hasler, P., “Characteristics and programming of floating-gate pfet switches in an fpaa crossbar network,” *IEEE International Symposium on Circuits and Systems (ISCAS) (IEEE Cat. No. 05CH37618)*, vol. Vol. 1, pp. 468 – 71, 2005.

[7] Hall, T. S., Twigg, C. M., Gray, J. D., Hasler, P., and Anderson, D. V., “Large-scale field-programmable analog arrays for analog signal processing,” *IEEE Transactions on Circuits and Systems*, 2004.

[8] Hall, T. S., Twigg, C. M., Hasler, P., and Anderson, D. V., “Application performance of elements in a floating–gate FPAA,” in *Proceedings of the International Symposium on Circuits and Systems*, pp. 589–592, 2004.

[9] Hall, T. S., Twigg, C. M., Hasler, P., and Anderson, D. V., “Developing large–scale field–programmable analog arrays,” in *Proceedings of the International Parallel and Distributed Processing Symposium*, pp. 142–147, 2004.

[10] Hasler, P. and Lande, T. S., “Overview of floating-gate devices, circuits, and systems,” 2001.

[11] Hasler, P., Minch, B., and Diorio, C., “Adaptive circuits using pfet floating-gate devices,” *Proceedings 20th Anniversary Conference on Advanced Research in VLSI*, pp. 215 – 29, 1999.

[12] KUB, F. J., MOON, K. K., MACK, I. A., and LONG, F. M., “Programmable analog vector matrix multipliers,” *IEEE JOURNAL OF SOLID-STATE CIRCUITS*, vol. 25, no. 1, pp. 207–214, 1990.

[13] LEE, K. and GULAK, P., “A transconductor-based field-programmable analog array,” in *IEEE International Solid–State Conference Digest of Technical Papers*, pp. 198–199, feb 1995.

[14] LOOBY, C. A. and LYDEN, C., “Op–amp based cmos field–programmable analogue array,” in *IEE Proceedings Circuits, Devices and Systems*, vol. 147, pp. 93–99, apr 2000.

[15] MINCH, B. A., *Analysis, Synthesis, and Implementation of Networks of Multiple-Input Translinear Elements*. PhD thesis, California Institute of Technology, May 1997.

[16] MINCH, B. A., “Synthesis of static and dynamic multiple-input translinear element networks,” *IEEE Transactions on Circuits and Systems*, vol. 51, pp. 409–421, Feb. 2004.

[17] MINCH, B. A., DIORIO, C., HASLER, P., and MEAD, C. A., “Translinear circuits using subthreshold floating-gate mos transistors,” *ANALOG INTEGRATED CIRCUITS AND SIGNAL PROCESSING*, vol. 9, no. 2, pp. 167–179, 1996.

[18] MULDER, J., SERDIJN, W., VAN DER WOERD, A., and VAN ROERMUND, A., “A generalized class of dynamic translinear circuits,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 48, pp. 501 – 4, May 2001.

[19] NOZAMA, H. and KOKYAMA, S., “A thermionic electron emission model for charge retention in SAMOS structures,” *Japanese Journal of Applied Physics*, vol. 21, pp. L111–L112, Feb. 1992.

[20] PREMONT, C., GRISEL, R., ABOUCHI, N., and CHANTE, J.-P., “Current–conveyor based field programmable analog array,” in *IEEE Midwest Symposium on Circuits and Systems*, vol. 1, pp. 155–157, aug 1996.

[21] RAMIREZ-ANGULO, J. and LOPEZ, A. J., “Mite circuits: The continuous-time counterpart to switched-capacitor circuits,” *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II-ANALOG AND DIGITAL SIGNAL PROCESSING*, vol. 48, no. 1, pp. 45–55, 2001.

[22] RAMIREZANGULO, J., CHOI, S. C., and GONZALEZALTAMIRANO, G., “Low-voltage circuits building-blocks using multiple-input floating-gate transistors,” 1995.

[23] SCHWARTZ, D. B., HOWARD, R. E., and HUBBARD, W. E., “A programmable analog neural network chip,” *IEEE JOURNAL OF SOLID-STATE CIRCUITS*, vol. 24, no. 2, pp. 313–319, 1989.

[24] Serrano, G., Smith, P., Lo, H., Chawla, R., Hall, T., Twigg, C., and Hasler, P., “Automatic rapid programming of large arrays of floating-gate elements,” *2004 IEEE International Symposium on Circuits and Systems (IEEE Cat. No.04CH37512)*, vol. Vol.1, pp. 373 – 6, 2004.

[25] SHIBATA, T. and OHMI, T., “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.

[26] Smith, P. D., Kucic, M., and Hasler, P., “Accurate programming of analog floating-gate arrays,” in *Proceedings of the International Symposium on Circuits and Systems*, pp. 489–492, 2002.

[27] Srinivasan, V., Serrano, G., Gray, J., and Hasler, P., “Precision cmos amplifier using floating-gate offset cancellation,” *IEEE Custom Integrated Circuits Conference*, pp. 739–743, 2005.

[28] Subramanian, S., Anderson, D., and Hasler, P., “Synthesis of static multiple input multiple output mite networks,” *IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 1, pp. I–189– I–192, 2004.

[29] Sun, Y., ed., *Design of high frequency integrated analogue filters*. London, UK: The Institution of Electrical Engineers, 2002.

[30] Takeda, E., Shimizu, A., and Hagiwara, T., “Role of hot-hole injection in hot-carrier effects and the small degraded channel region in mosfet’s,” *IEEE Electron Device Letters*, vol. 4, no. 9, pp. 329–331, 1983.

[31] Taur, Y. and Ning, T. H., *Fundamentals of modern VLSI devices*. New York, NY, USA: Cambridge University Press, 1998.