NEURAL DYNAMICS IN RECONFIGURABLE SILICON

A Thesis
Presented to
The Academic Faculty

by

Arindam Basu

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

Georgia Institute of Technology
May 2010

NEURAL DYNAMICS IN RECONFIGURABLE SILICON

Approved by:

Assoc. Professor Paul Hasler
(Advisor), Committee Chair
School of Electrical and Computer
Engineering
Georgia Institute of Technology

Asst. Professor Christopher Rozell
School of Electrical and Computer
Engineering
Georgia Institute of Technology

Assoc. Professor Robert Butera
School of Electrical and Computer
Engineering
Georgia Institute of Technology

Professor Avis Cohen
Department of Biology
University of Maryland

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

Professor Steven Baer
Department of Mathematics
Arizona State University

Date Approved: February 22, 2010

To my parents and my sister,

the reasons for my existence.

ACKNOWLEDGEMENTS

Finally I am at a point where I am writing the acknowledgement section of my dissertation! And I have a lot of people to thank for getting me here. First, this Ph.D. would not have been possible without the constant help and support from my family - in particular, my parents and sister. I sincerely thank you for your patience and hope I can live up to your expectations. Second, my friends in Atlanta have been a wonderful support system. Doing a Ph.D., in case you did not know, is an extremely demanding task - and you need all the emotional support you can get. So I would like to thank Padmanava Sen, Saikat Sarkar, Arup Polley, Atri Dutta, Mrinmoy Ghosh, Bevin Perumana, Prabir Saha, Subho Chatterjee, Payel Paul, Kaushik Kundu, Priya Kundu, Nand Kishore, Neha Jha, Rajarshi Mukhopadhyay, Rachana Mukhopadhyay, Jithu Susan Mathew, Rita Polley, Ajay Kumar, Bhaskar Banerjee, Debrup Das, Shreyas Sen, Zannatul Ferdous, Debesh Bhatta, Tonmoy Mukherjee, Tapobrata Bandyopadhyay, Chandan Dasgupta, Ananda Barua, Aritra Banerjee, Ayan Chakrabarti and Payel Chakrabarti. I will never forget the potlucks, tennis, soccer and cricket games, movies or pure 'bhat' sessions we had - probably the best few years of my life.

In Georgia Tech, I have been extremely fortunate to have an advisor who is brilliant in a wide variety of technical areas spanning circuits, devices, mathematics and neuroscience. I have been deeply influenced by his research and hope to continue collaborating on many fronts. And thanks a lot Paul for introducing me to non-linear dynamics - that will probably be a research focus for me throughout my life.

I also enjoyed working with Dr. Degertekin on sensor interfacing and Dr. Butera on neural dynamics and electrophysiology. Here, I have to mention the help I got from Dr. Shui-Nee Chow and Dr. Predrag Cvitanovic in understanding the nuances

of differential equations and chaos.

Special thanks are due to my labmates in ICELAB - Stephen, Scott, Craig, Csaba, Bo, Jordan, Shubha, Degs, Richie, Farhan, Kofi, Guillermo, Venkatesh, Ken, Brian, Ryan, Thomas, Shakeel, Gokce; will cherish the night long chip designs as well as the poker sessions. In particular, I enjoyed the various stimulating discussions I have had with Stephen throughout my tenure at GaTech. It was a pleasure overlapping with you and I wish you all the best for your future.

Last but not the least, I wish to thank Janet, Daphne, Nydia and the ECE graduate office for helping through all the administrative hurdles during my Ph.D.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

This work is a first step towards a long-term goal of understanding computations occurring in the brain and using those principles to make more efficient machines. The traditional computing paradigm calls for using digital supercomputers to simulate large scale brain-like neural networks resulting in large power consumption which limits scalability or model detail. For example, IBM's digital simulation of a cat brain with simplistic neurons and synapses consumes power equivalent to that of a thousand houses! Instead of digital methods, this work uses analog processing concepts to develop scalable, low-power silicon models of neurons which have been shown to be around ten thousand times more power efficient. This has been achieved by modeling the *dynamical behavior* of Hodgkin-Huxley (H-H) or Morris-Lecar type equations instead of modeling the exact equations themselves. In particular, the two silicon neuron designs described exhibit a Hopf and a saddle-node bifurcation. Conditions for the bifurcations allow the identification of correct biasing regimes for the neurons. Also, since the hardware neurons compute in real time, they can be used for dynamic clamp protocols in addition to computational experiments.

To empower this analog implementation with the flexibility of a digital simulation, a family of field programmable analog array (FPAA) architectures have been developed in 0.35 μm CMOS that provide reconfigurability in the network of neurons as well as tunability of individual neuron parameters. This programmability is obtained using floating-gate (FG) transistors. The neurons are organized in blocks called computational analog blocks (CAB) which are embedded in a programmable switch matrix. An unique feature of the architecture is that the switches, being FG

elements, can be used also for computation leading to more than 50,000 analog parameters in 9 sq. mm. Several neural systems including central pattern generators and coincidence detectors are demonstrated. Also, a separate chip that is capable of implementing signal processing algorithms has been designed by modifying the CAB elements to include transconductors, multipliers etc. Several systems including an AM demodulator and a speech processor are presented.

An important contribution of this work is developing an architecture for programming the FG elements over a wide dynamic range of currents. An adaptive logarithmic transimpedance amplifier is used for this purpose. This design provides a general solution for wide dynamic range current measurement with a low power dissipation and has been used in imaging chips too.

A new generation of integrated circuits have also been designed that are 25 sq. mm in area and contain several new features including adaptive synapses and support for smart sensors. These designs and the previous ones should allow prototyping and rapid development of several neurally inspired systems and pave the path for the design of larger and more complex brain like adaptive neural networks.

CHAPTER I

WHY RECONFIGURABLE SILICON ?

Pioneering work in modeling the dynamics of a biological neuron was done by Hodgkin and Huxley in the 1950's. Since then mathematical models of varying complexities have been proposed and studied, facilitated largely by the rapid growth of digital computers. However, simulating a large number of these "mathematical" neurons is still a daunting task, especially when there are multiple time scales involved in the differential equations (stiff equations). For example, the Hodgkin-Huxley (HH) equations for dynamics of one sodium and one potassium channel are 'stiff' for slow inactivation and fast activation of sodium conductance. Hence, there is ample reason to search for dedicated silicon implementations that efficiently perform this computation. Among the possible silicon implementations, the analog one is preferable over the digital one for several reasons. Firstly, since the biological system being modeled is a continuous-time one, an analog implementation is naturally better suited because it avoids errors due to discretization, and possible convergence issues. Secondly, with the progress in semiconductor technology, power supply levels have become similar to biology opening a new possibility of the same silicon model to interface with live neurons [70] [56]. In this context, the analog implementation has the possibility of being much smaller, and less power-hungry than the digital, making it also amenable for implants. Hence, there has been a trend towards implementing the modeling equations on an analog platform, the earliest instance of which dates back to Richard Fitzhugh in the 1960's. However, the true power of a silicon neuron implementation is not utilized unless arbitrary networks composed of these neurons can be constructed

at will. This typically requires fabrication of new IC chips that is a costly, time consuming and error-prone process. Also, one IC has at most a few possible networks. The solution to this problem is to fabricate a reconfigurable silicon neural system that has the capability of creating arbitrary interconnections. Going one step further, it would be even more useful to have a system that can be programmed on the fly based on external signals. This would provide an efficient means of implementing feedback in neural systems based on signals from both the external world and other neurons.

In this research, a framework for generating silicon neuron models with desired dynamics and their integration into a reconfigurable IC is explored. Silicon channel models, much more compact and power efficient than earlier ones, are developed by using bifurcations of the traditional neuron models as a key. A Field Programmable Analog Array (FPAA) type architecture for exploring neural dynamics of different networks is also developed. The goal of this research is to develop a dynamical systems approach for creating silicon neuron models and to architect a system of neurons where networks of arbitrary connections can be made and modified based on external or feedback signals.

The impact of this work will be observable at many levels. Firstly, the framework for reconfigurable systems that is developed can be easily modified for other applications like analog signal processing or sensor interfacing. Secondly, this framework allows one to study a number of different aspects of neural systems. While the detailed channel models allow mimicking the real biophysics of nerve cells, one can also implement simpler neuron models and modify the interconnections to study the importance of the topology of the network. This architecture also allows one to explore answers to the question "How important is the dynamics of individual cells to the behavior of the network?" Solving this question in hardware is more useful as the presence of noise naturally eliminates the solutions that are not robust (theoretically analyzing the situation needs knowledge of stochastic differential equations, which

are intractable if the network is anything beyond a few neurons). Also, using floating gates for real-time adaptation enables realistic time constants on the order of tens of seconds. This will allow systems to interact with real neurons in a closed loop. Thus, it is obvious that this work has the potential to open up several avenues for research and enable scientists to explore many interesting questions.

This thesis is organized into nine sections, six of which have each resulted in at least one conference publication and at least one journal publication (published or in review). Sections II and III describe two silicon neurons that exhibit two different co-dimension one bifurcations - Hopf and Saddle-node. Conditions for biasing these circuits in the correct regime are developed and non-linear dynamical analysis of their properties are presented. This unique methodology of bifurcation-based silicon neuron design has resulted in compact designs with one of the lowest reported power dissipation. Section IV details a field-programmable analog array (FPAA) architecture for integrating these neuron designs into a larger network with flexible topology and programmable interconnection (synaptic) strength. The FPAA architecture developed is general and can be used for other applications by replacing the neuron circuits with other elements (transistors, amplifiers etc). Section V describes some larger neural systems compiled on an FPAA for spiking neural networks while section VI presents an FPAA for analog signal processing applications. In section VII, a more detailed description of a floating-gate (FG) device (the circuit element responsible for this reconfigurability) and an architecture to modify the charge on its gate is presented. Section VIII details one particular circuit in the FG programming architecture that is essential for measuring the current in the FG device. The logarithmic current to voltage converter developed in this work utilizes adaptation to reduce power consumption and has been used in other designs such as imagers. Finally, some of the ongoing and future work regarding smart sensors and run-time adaptation in FPAAs is presented in section IX along with conclusions.

CHAPTER II

A HOPF NEURON

In this chapter, a silicon neuron that exhibits Hodgkin-Huxley type dynamics is modeled. As mentioned earlier, there have been multiple instances of modeling Hodgkin-Huxley equations in silicon. Almost all of the implementations use the Hodgkin-Huxley (H-H) formalism, some of them using a set of equations simpler than the original H-H equations [18] [77] while others have modeled in detail the full set of equations [69] [95] [2] [40] [96].

The silicon neuron presented here uses a MOSFET to represent a channel [37] since carrier transport phenomenon are similar in both. Different control amplifiers are used to control the gate of these "channel" transistors so that the voltage clamp responses of the individual channels resemble biology. A combination of one sodium and one potassium channel makes a basic neuron as shown in Fig. 1. While [37] demonstrated voltage-clamp data similar to biology, we present analytical and experimental bifurcation diagrams with varying input current that matches biology and the H-H model. Also the analytical framework developed in this chapter allow easy identification of valid biasing regimes of the circuit. Instead of proving the topological equivalence to H-H equations, we concentrate on the bifurcations exhibited by this circuit and use it as a metric to validate the efficacy of this model. An introduction to this approach was presented in [13] where biasing conditions for Hopf bifurcations were explored. This chapter presents a complete analysis of the center-manifold reduction, bifurcation diagrams based on continuation, a description of the designed chip, modifications to the circuit for independent parameter control, and measurement results from a silicon prototype.

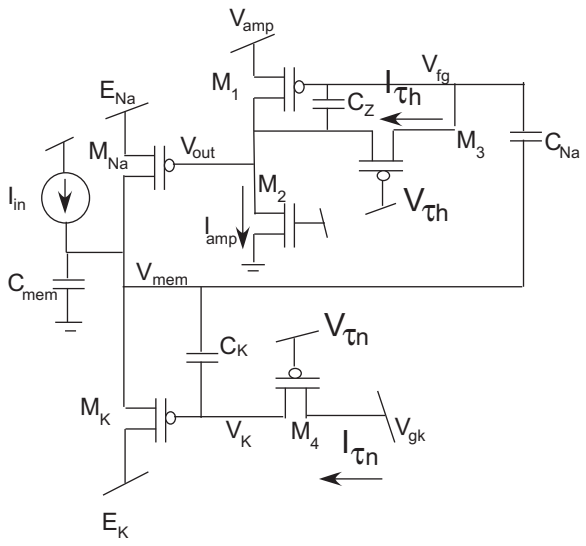


Figure 1: **A Neuron with two channels:** A simple neuron with one sodium channel (M_{Na}) and one potassium channel (M_K). The sodium channel transistor is controlled by an amplifier with bandpass characteristics giving it the fast activation and slow inactivation dynamics. The potassium channel transistor is controlled by a low-pass amplifier providing it with slow activation.

The approach of studying bifurcations is useful because it is believed that computational properties of neurons are based on the bifurcations exhibited by these dynamical systems in response to some changing stimulus [53] [87]. Thus, it is likely that all models which present the same set of bifurcations should be equally good in analyzing and modeling neurons. For example, any neuron exhibiting a Hopf bifurcation can easily signal when a stimulus crosses a threshold by initiating a spike-train, while those exhibiting saddle node bifurcations can encode the strength of a stimulus in their firing rate. Hence, by showing that this silicon neuron has bifurcations similar to a certain class of biological neurons, we can claim that the silicon neuron can also perform similar computations.

A similar approach has been employed in [60] but they do not explore the property of excitation block. In fact, we propose an alternate dynamic explanation of the excitation block property. Contrary to the supercritical Hopf bifurcation that causes this property in the traditional H-H model, our model shows a subcritical Hopf bifurcation and a fold bifurcation with decreasing limit cycle amplitude. Experimentally,

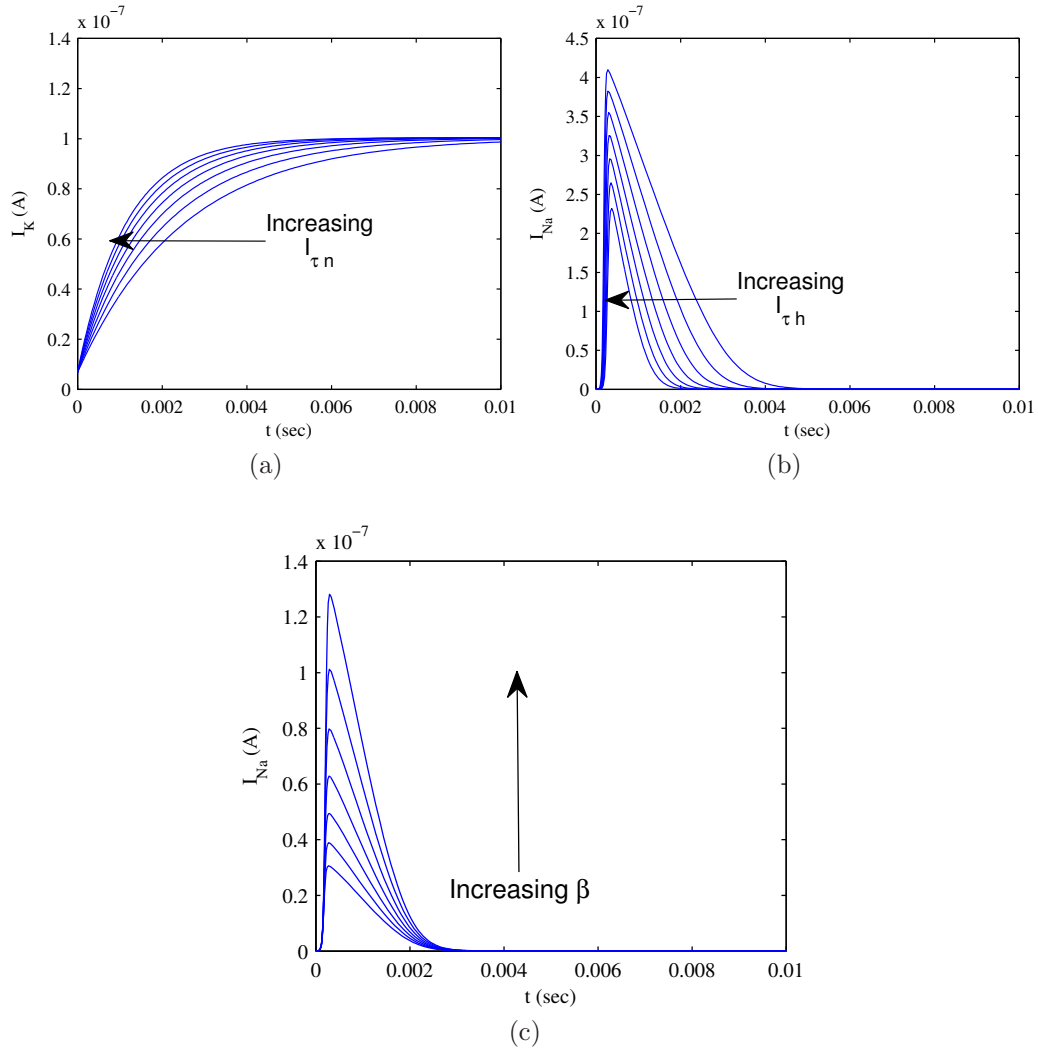


Figure 2: **Relation of parameters to biology:** Simulations of voltage clamps of the numerical model to show relationship of the model parameters to biological properties of the neuron. In each case, a voltage step of 75 mV was applied for different parameter values. (a) Effect of increasing $I_{\tau n}$ is to make the activation of potassium faster. (b) Increasing $I_{\tau h}$ has the effect of increasing inactivation of sodium current which reduces the maximum sodium current and increases the speed of inactivation. (c) Increasing β increases the maximum value of sodium current during an action potential.

it is impossible to distinguish the two mechanisms due to ambient noise which pushes the solution into the basin of attraction of the decreasing amplitude limit cycle and thus the bifurcation appears to be a supercritical Hopf. The details of the circuit and its dynamics are described in the following sections.

2.1 Circuit Operation and Model

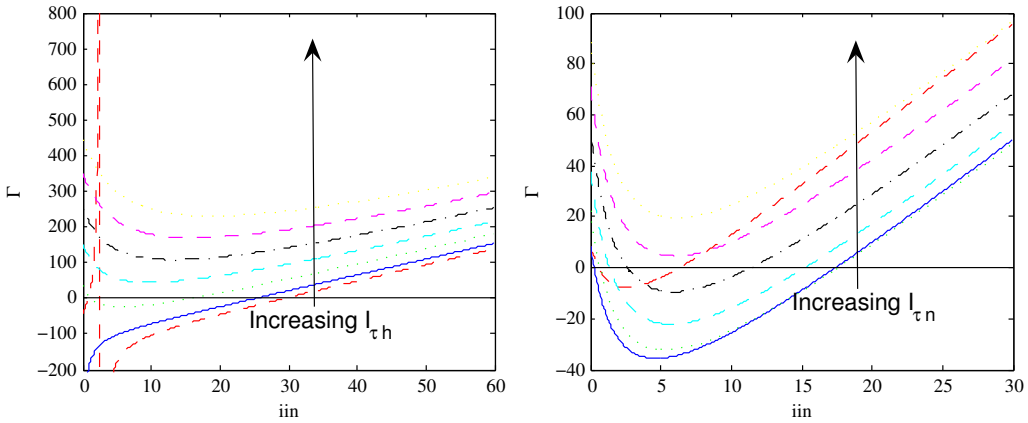
In this section, the operation of the circuit is discussed and a mathematical model for its analysis is developed. We also mention the relation between the model parameters and biological quantities. As shown in Fig. 1, the channels consist of a channel transistor and a control amplifier. The membrane voltage V_{mem} varies between E_{Na} and E_K (similar to sodium and potassium reversal potentials) which have a difference of $200mV$ that is close to biology. The sodium amplifier is bandpass (since both activation and inactivation are present [37]) with a gain set to approximately -8 by the ratio of the capacitors C_{Na} and C_K. I_{amp} (current in M2) is responsible for the activation of the sodium channel and $I_{\tau h}$ (current in M3) provides inactivation. M3 acts as a short between the nodes V_{out} and V_{fg} at DC and helps set their equilibrium values. In some sense, it acts as a non-linear resistor in feedback across the amplifier. The power supply for the sodium amplifier, V_{amp} along with the bias current I_{amp} sets the DC operating point for the node V_{out}. This in turn fixes a bias current, I_{Na0} through M_{Na}. The potassium channel is controlled by a lowpass amplifier (since it is only activating) with the filter's corner frequency being set by $I_{\tau n}$, the current through M4. M4, like M3 acts as a non-linear resistance. Other non-linear resistors like an NMOS or a diode connected MOS can also be used and are similar in the small-signal sense but give rise to different large-signal dynamics. Interestingly, all those non-linear resistors share the same linearized form and hence the existence of a Hopf bifurcation, which is only dependent on the Jacobian, is maintained in all implementations. The only difference is in the type of Hopf bifurcation due to the higher order terms in the normal form. However, we shall not discuss more about that in this chapter. The channel currents from M_{Na} and M_K are added on the membrane capacitor, C_{mem} on the node V_{mem}. An external stimulus current, I_{in} is considered as a bifurcation parameter.

The generation of an action potential consists of the following phases: a small

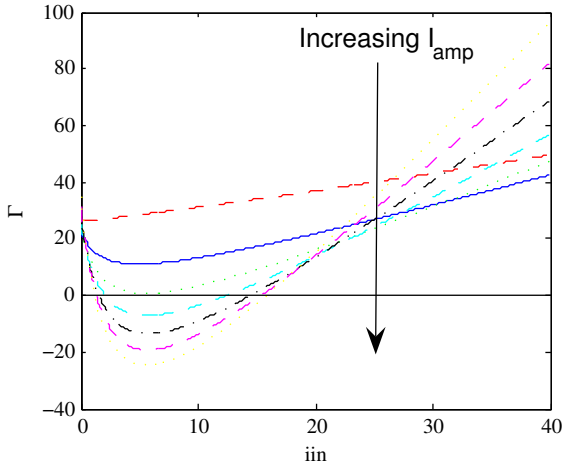
positive perturbation on V_{mem} causes the sodium amplifier to pull down the gate of M_{Na} if the perturbation was fast enough. This leads to an increase in the sodium channel current. The perturbation also couples onto the gate of M_K through the capacitor C_K and thus the increase in potassium current is limited. Hence the net positive current increases V_{mem} more by positive feedback. This is the upstroke of the action potential leading to depolarization of the membrane. After some time, the sodium current decreases because of the ohmic effect of M_{Na}, saturation of output of sodium control amplifier and eventual pull-up by $I_{\tau h}$. Also, the potassium current increases due to discharge of gate of M_K by $I_{\tau n}$. The decrease of sodium and increase of potassium currents lead to a decrease in V_{mem} which is again boosted by the positive feedback of the sodium control. This change also couples onto the gate of M_K pushing it below equilibrium value leading to hyperpolarization of the membrane which is then eventually pulled up due to recovery by $I_{\tau n}$. We now develop the dynamical equations for the system by using KCL at the V_{mem}, V_k, V_{fg} and V_{out} nodes respectively. The variables x, n, h and m are defined as follows:

$$\begin{aligned} x &= \frac{\delta V_{mem}}{U_T}; n = \frac{\delta V_k}{U_T} \\ h &= \frac{\delta V_{fg}}{U_T}; m = \frac{\delta V_{out}}{U_T}, \end{aligned} \tag{1}$$

where the δV variables represent changes in the voltages from steady state and U_T is thermal voltage. Notice that there are four variables as in the H-H equations which is natural as both of them model the same system.



(a) (b)



(c)

Figure 3: **Effect of varying parameters:** Condition for a Hopf bifurcation is a pair of imaginary eigen-values with zero real part. The zero crossings of Γ in eq. 10 provide necessary conditions for such a case when (a) $I_{\tau h}$ is varied (b) $I_{\tau n}$ is varied and (c) I_{amp} is varied. We want to bias the neuron such that we get at least two such intersections which could lead to two Hopf bifurcations. From the figures we see that if I_{amp} is large enough it does not affect the intersections much. This is what you would expect as the "m" variable in H-H equations is considered much faster than "n" or "h" in all analysis.

$$
\begin{aligned}
U_T\dot{x} &= \frac{I_{Na0}}{C_{mem}}e^{-m}\{1-(1+\frac{I_{IN}}{I_{Na0}})e^{x-4}\}+\frac{I_{\text{in}}}{C_{mem}} \\
&\quad -\frac{I_{K0}}{C_{mem}}e^{x-n} \\
U_T\dot{n} &= U_T\dot{x}-\frac{I_{\tau_n}}{C_K}f(n) \\
U_T\dot{h} &= U_T\dot{x}+\frac{I_{amp}}{8C_Z}\{e^{-h}-1+e^{-m-\beta}\} \\
U_T\dot{m} &= U_T\dot{x}+\frac{1.125I_{amp}}{C_Z}\{e^{-h}-1+e^{-m-\beta}\}+\frac{I_{\tau_h}}{C_Z}g(h,m)
\end{aligned} \tag{2}
$$

where f and g represent the currents through M4 and M3 respectively. This generalization allows for analyzing a general non-linear resistor in that place. Here, the functions are considered to be $f = e^n - 1$ and $g = e^h - e^m$ since one terminal of M4 is at a fixed potential while both terminals of M3 can vary. In this equation, I_{IN} is the average or DC component of I_{in}, the total input current. Also, note the term $1 + \frac{I_{IN}}{I_{Na0}} e^{x-4}$ in the first equation which models the sodium transistor entering ohmic region which is one essential term for the excitation block property described later. To derive this term, we assume that M_k is in saturation to obtain an expression for V_{mem}, the deviation of the DC voltage at the membrane from its value V_{mem0} when $I_{IN} = 0$:

$$\begin{aligned} I_{Na0} + I_{IN} &= I_{Na0} e^{V_{mem}/U_T} \\ \Rightarrow V_{mem} &= U_T log(1 + \frac{IN}{I_{Na0}}) \end{aligned} \tag{3}$$

where I_{Na0} is the bias current in the sodium channel corresponding to V_{mem0}. The variable x represents changes from this DC level defined by V_{mem}. Now, assuming for simplicity that κ, the coupling of the gate onto the channel of the transistor is 1, the equation for the sodium current may be derived as:

$$\begin{aligned} I_{Na} &= I_0 e^{-\frac{V_{out}}{U_T}} (e^{\frac{E_{Na}}{U_T}} - e^{\frac{V_d}{U_T}}) \\ &= I_{Na0} e^{-m} (1 - e^{\frac{-E_{Na} + V_{mem0} + V_{mem} + \delta V mem}{U_T}}) \end{aligned} \tag{4}$$

where I_{Na0} is the DC current in the sodium channel ignoring the effect of the drain of the transistor and V_d is the drain potential of M_{Na}. Now assuming that at $I_{\text{in}} = 0$, $E_{Na} - V_{mem0} = 4U_T$, and using 3, we get:

$$\begin{aligned} I_{Na} &= I_{Na0} e^{-m} (1 - e^{\frac{V_{mem}}{U_T}} e^{x-4}) \\ &= I_{Na0} e^{-m} (1 - (1 + \frac{I_{IN}}{I_{Na0}}) e^{x-4}) \end{aligned} \tag{5}$$

Since we are concerned with only static bifurcations of the equilibrium, we consider only the case where the input current is DC, i.e. $I_{\text{in}} = I_{IN}$. Henceforth, only the term

I_{in} is used in the chapter. It is also interesting to note that the voltages in this circuit are normalized to the thermal voltage. This seems very natural given the fact that the sodium and potassium potentials are also proportional to the thermal voltage times logarithm of ratio of ionic concentrations. Ohmic effects of M_K and $M1$ are ignored for this particular case but they are modeled for M2 through the parameter β. These are important to determine the maximum sodium conductance during an action potential.

The effect of these mathematical parameters on the biological function of the neuron can be understood by considering their effects on voltage clamp experiments. As an example, Fig. 2 shows a simulation of a voltage clamp for a voltage step of 75 mV with different parameter values in eq. 2. The nominal parameter values for this simulation are $I_{\tau h} = 8pA, I_{\tau n} = 8pA, I_{amp} = 150pA, Cz = .08pF, Ck = 0.5pF, \beta = 10, U_T = 0.025, I_{K0} = 5nA, I_{Na0} = 0.05nA$. In Fig. 2 (a), $I_{\tau n}$ is varied from $0.5 - 1.1pA$ (keeping the others at their nominal values) showing an increased speed of activation of potassium channel. In Fig. 2 (b), $I_{\tau h}$ is varied from 0.5 - 1.1 pA leading to faster inactivation of sodium channels. Finally, 2 (c) shows that increasing β increases the maximum sodium conductance during an action potential. Among the other parameters, increasing I_{amp} increases the speed of sodium activation, while I_{Na0} and I_{K0} determine the baseline current levels in the channels at rest.

To reduce the number of parameters, we do another re-normalization as follows:

$$\begin{aligned} \tau &= \frac{tI_{Na}}{C_{mem}U_T} \\ C_i' &= \frac{C_i}{C_{mem}} \\ I_j' &= \frac{I_j}{I_{Na0}} \\ y &= n - x, z = h - x, w = m - x, \end{aligned} \tag{6}$$

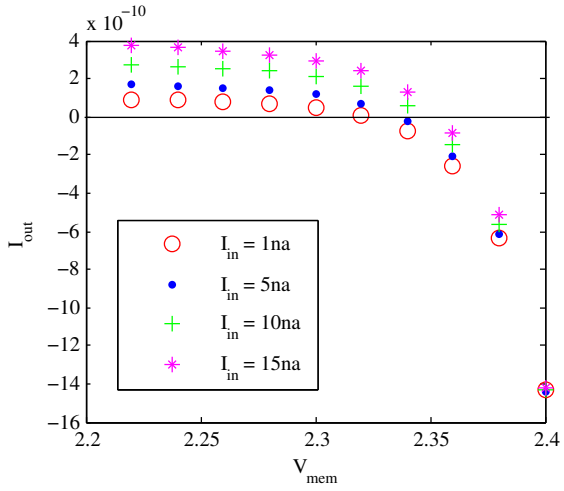


Figure 4: **Measured I-V characteristic:** The I-V characteristic of the neuron shows monotonic behavior and thus has one fixed point. The equilibrium voltage increases with increasing I_{in}.

and then dropping the primes for economy of notation we get:

$$\begin{aligned}
\dot{x} &= e^{-(w+x)}\{1-(1+I_{\text{in}})e^{x-4}\}+I_{\text{in}}-I_{K0}e^{-y} \\
\dot{y} &= -\frac{I_{\tau_n}}{C_K}\{e^{y+x}-1\} \\
\dot{z} &= \frac{I_{amp}}{8C_Z}\{e^{-(z+x)}-1+e^{-(w+x+\beta)}\} \\
\dot{w} &= \frac{1.125I_{amp}}{C_Z}\{e^{-(z+x)}-1+e^{-(w+x+\beta)}\}+ \\
&\frac{I_{\tau_h}}{C_Z}\{e^{z+x}-e^{w+x}\}
\end{aligned} \tag{7}$$

where $I_{K0}=(1+I_{\text{in}})(1-e^{-4})$ includes the effect of input stimulus.

2.1.1 Finding Fixed Points

The first piece of information one desires to know about a dynamical system are its attractors and invariant sets, the simplest of which are its equilibria or fixed points. The fixed points of (7) are found by setting the RHS of the equations to zero which gives the origin as an unique fixed point. An easy way of finding the equilibria is to exploit the fact that a fixed point corresponds to zero current in all the capacitors. Thus, we need to sweep V_{mem} and find the steady state current through the voltage

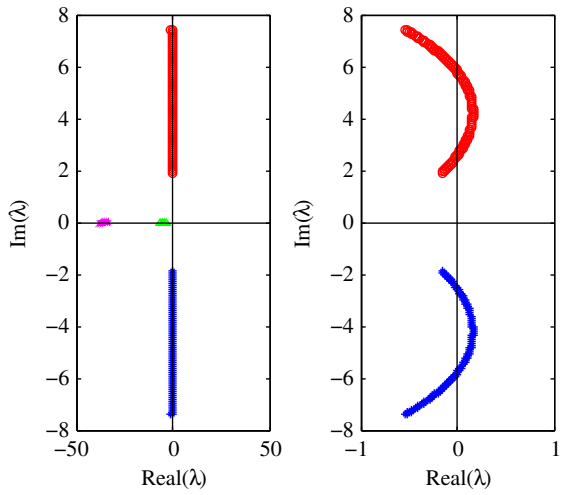


Figure 5: **Spectrum of the stability matrix:** The eigen-values of the stability matrix are plotted with increasing I_{in} for $I_{\tau_h} = I_{\tau n} = 0.8, I_{amp} = 15$. The figure on the right shows a closeup of the complex eigen-values demonstrating two Hopf bifurcations.

source. The number of zero crossings of this I-V curve give the number of equilibria. Fig 4 shows measured I-V characteristic. The I-V characteristic is monotonic since at steady state, the voltage at the gates of M_{Na} and M_K are constant, reducing it to the I-V curve of a source follower. It should be noted that the increase of resting membrane potential with increasing stimulus current that can be seen in Fig. 4 is not obtained theoretically because of the way we define I_{K0}. Finding the I-V characteristic is an useful method in general for finding equilibria of an unknown circuit, especially when there are multiple equilibria (this neuron has multiple equilibria in the limit $I_{\tau h} \to 0$).

2.1.2 Conditions for Hopf Bifurcations

Since Class 2 excitability of neurons is defined by spontaneous firing with large amplitude and non-zero frequency in response to an input current stimulus (that is larger than a certain critical value), it can be related to subcritical Hopf bifurcations. Hence, to prove that the silicon neuron possesses similar dynamical properties, we need to find conditions for the existence of a similar bifurcation. In addition, the limit cycle amplitude reduces with increasing I_{in} and the equilibrium regains stability through

a Hopf bifurcation which may be subcritical or supercritical. This phenomenon is termed excitation block and observed in layer 5 pyramidal neuron of rat's visual cortex [55].

The relevant conditions for Hopf bifurcation in our case may be stated as follows:
(1) Two of the eigen-values of the stability matrix of the equilibrium must be complex conjugates with zero real part and non-zero imaginary part at the bifurcation.
(2) The derivative of the real part of the complex eigen-values with respect to the parameter must be non-zero at the bifurcation [44]. In our case, we want the resting state to be stable before the bifurcation. Hence, the two real eigen-values not involved in the bifurcation will be negative. Using these conditions, we want to find the region of parameter space where our circuit can produce two Hopf bifurcations with increasing I_{in}.

To use the above conditions, we equate the characteristic polynomial of the stability matrix of the equilibrium to the desired form with two real and two complex eigen-values with zero real part at equilibrium:

$$x^4 + a_3x^3 + a_2x^2 + a_1x + a_0 = (x^2 + \omega_0^2)(x^2 + (\lambda_1 + \lambda_2)x + \lambda_1\lambda_2), \tag{8}$$

where $-\lambda_1$ and $-\lambda_2$ are the real eigen-values.The variables (a_0, a_1, a_2 and a_3) can be solved as follows:

$$\begin{aligned} a_3 &= \lambda_1 + \lambda_2 \\ a_2 &= \lambda_1\lambda_2 + \omega_0^2 \\ a_1 &= \omega_0^2(\lambda_1 + \lambda_2) \\ a_0 &= \omega_0^2\lambda_1\lambda_2 \end{aligned} \tag{9}$$

Then the conditions for Hopf bifurcation become:

$$\begin{aligned} a_3 &= \lambda_1 + \lambda_2 > 0 \\ \frac{a_1}{a_3} &= \omega_0^2 > 0 \\ \Gamma &= a_2 - \frac{a_1}{a_3} + \frac{a_0 a_3}{a_1} = 0 \end{aligned} \tag{10}$$

These conditions are necessary but not sufficient since the condition for non-zero derivative of the real part of the complex eigen-values needs to be appended. The condition on Γ exists as the four variables a_0, a_1, a_2 and a_3 are each expressed in terms of lesser number of parameters (λ_1, λ_2 and ω_0). Fig. 3 shows plots of Γ with varying I_{in} for different $I_{\tau h}, I_{\tau n}$ and I_{amp}. The zero-crossings of these curves along with the first two conditions in (10) give possible bifurcation points. In every case, we can see some sets of parameters do result in two zero crossings of Γ indicating possible valid biasing regimes. Choosing one such set of parameters ($I_{\tau_h} = I_{\tau n} = 0.8, I_{amp} = 15$), the eigen-values of the Jacobian were computed for different stimulus current values. Figure 5 depicts a plot of the eigen-values. The points where the curves cross the imaginary axis correspond to the two Hopf bifurcations with changing I_{in}).

2.2 Non-linear dynamics in the Neuron

In the previous section, we analyzed the conditions for Hopf bifurcations to determine the regions in the parameter space which lead to the possible existence of two Hopf bifurcations. In this section, the response of the system is studied when it is biased in such a desired regime.

To observe the bifurcations in the neuron circuit, a slowly increasing ramp of current was injected into the membrane in a SPICE simulation. It should be noted here that a slow ramp allows the system to stay in quasi-steady-state thus showing bifurcations of the equilibrium. On the other hand, pulse or step inputs have different properties which are discussed later. Fig. 6 shows the result of the SPICE simulation.

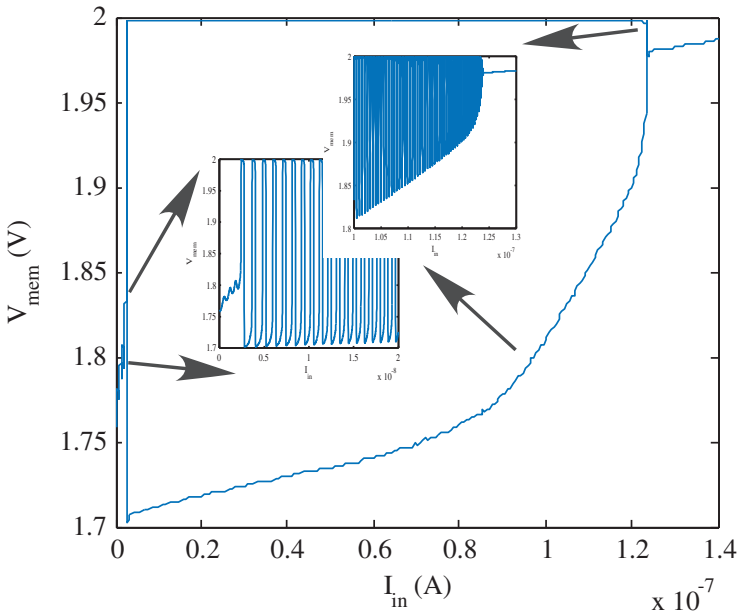


Figure 6: **Bifurcation in SPICE simulation:** Simulation of the neuron with I_{in} being a slowly increasing ramp. We can clearly see the loss of stability of the equilibrium and spontaneous oscillations at around $I_{\text{in}} = 2nA$. At around $I_{\text{in}} = 120nA$ the equilibrium becomes stable again. The closeup picture of the bifurcation around $I_{\text{in}} = 120nA$ shows the amplitude of oscillations gradually reducing.

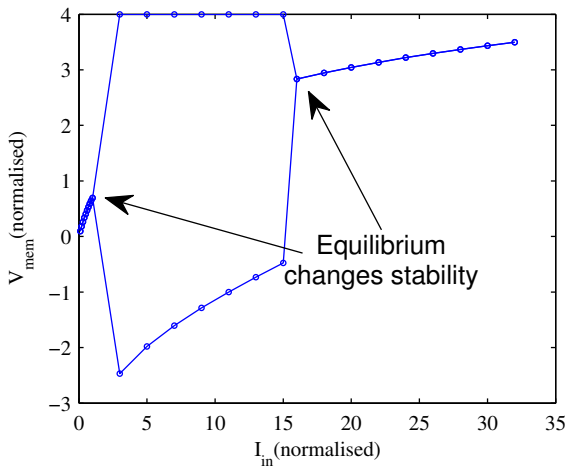


Figure 7: **Bifurcation in numerical integration:** The bifurcation of the theoretical model is observed by integrating the differential equations numerically. The equilibrium is perturbed to check its stability and the maximum and minimum of the resulting steady state is plotted. The nature of the obtained curve is similar to the one obtained from SPICE other than the apparent abrupt reduction in amplitude of the limit cycle at high currents.

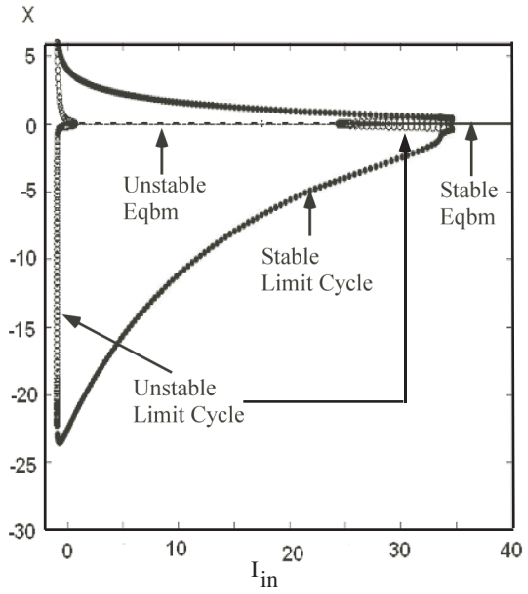


Figure 8: **Bifurcation diagram by Continuation:** The bifurcation of the theoretical model is observed by continuation using AUTO. Both the bifurcations are seen to be subcritical Hopf and the limit cycles appear and disappear by fold bifurcations. As the limit cycle amplitude reduces before the second fold bifurcation, it seems like a supercritical Hopf. The negative peak of the limit cycle is very large because the ohmic effect of transistor M_K has been neglected in the model equations as they are not essential for the bifurcations.

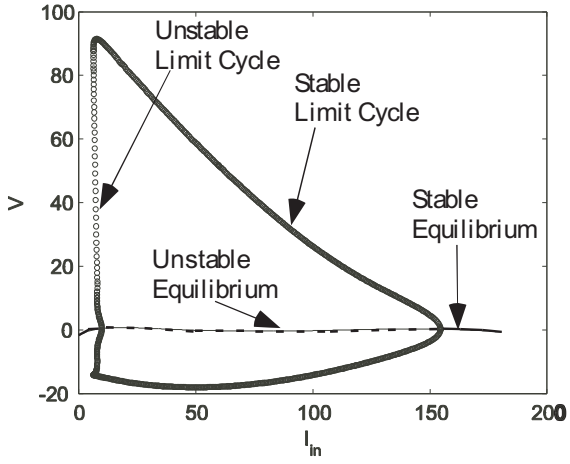


Figure 9: **Bifurcation diagram for HH model:** The Hopf bifurcation for smaller currents is subcritical. The limit cycle appears by a fold bifurcation and disappears by a supercritical Hopf. The reduction in amplitude before the subcritical Hopf is similar to Fig. 8.

The parameters for the simulation were $I_{\tau_h} = I_{\tau n} = 0.8, I_{amp} = 15$. The circuit exhibits spontaneous large oscillations when I_{in} is large enough, a classic case of Class 2 excitability. A zoomed picture near the critical value of I_{in} shows that right

after the equilibrium loses stability, large amplitude limit cycle solutions emerge. At a much larger value of the stimulus the equilibrium becomes stable again. This happens since the resting potential becomes close to E_{Na} making M_{Na} ohmic. From a small-signal perspective, the positive feedback loop of the sodium amplifier loses its gain in comparison to the negative feedback potassium channel and is not able to sustain oscillations. Thus, the ohmic regime of M_{Na} gives the behavior of the excitation block. A closeup of the oscillation near the bifurcation at higher currents shows the amplitude of the limit cycle gradually reducing before it disappears.

To verify that the mathematical model behaves in a similar manner compared to the circuit, the differential equations were numerically integrated to create a similar bifurcation diagram as shown in Fig. 7. To generate this diagram, the equilibrium for different I_{in} values was perturbed and the maximum and minimum of the resulting solution is plotted. It should be noted that the two bifurcation diagrams are different since, in Fig. 6 the limit cycle amplitude reduces before it collapses onto the equilibrium, while this is not visible in Fig. 7. From Fig. 7 the Hopf bifurcation at higher currents seems to be subcritical while it seems to be supercritical from the SPICE simulation. To rigorously understand the nature of the bifurcation, we project the system on a suspended center manifold [44] [31] at equilibrium corresponding to the higher current resulting in the normal form :

$$\begin{aligned} \dot{r} &= r(-0.03\mu - 0.006\mu^2 + (0.22 + 0.01\mu)r^2) \\ \dot{\theta} &= 6.5 + 0.1\mu + (-0.05 - 0.003\mu)r^2, \end{aligned} \tag{11}$$

where $\mu = I_{\text{in}} - I_{\text{in,b}}$ is the deviation of I_{in} from its critical value. The method for deriving this form is outlined in the appendix along with its application to the other Hopf bifurcation at lower currents. The cubic coefficient in the equation for amplitude is positive at bifurcation which indicates that the $r = 0$ solution is unstable at bifurcation and there exists an unstable limit cycle prior to bifurcation. This

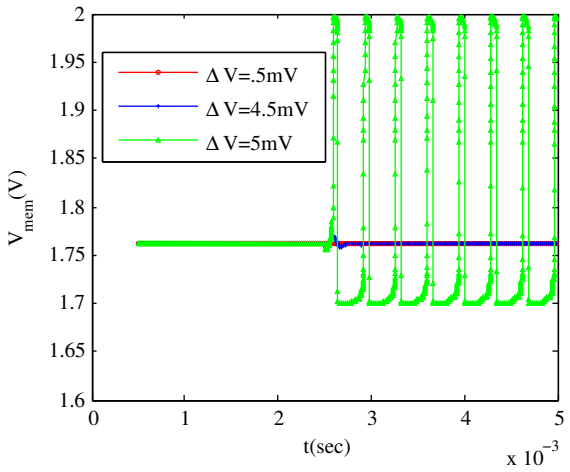


Figure 10: **Effect of Current impulses of different size:** Three current pulses of increasing amplitude were used to stimulate the neuron. While in first two cases the neuron returned to resting state, in the third case it burst into oscillations. This shows co-existence of a stable equilibrium and limit cycle.

conclusively shows that the bifurcation was indeed a subcritical Hopf.

To understand the reason for the apparent anomaly between the numerical results and the SPICE simulation, one needs to consider the bifurcations of the limit cycles separately. We used continuation of solutions in AUTO [35], a freely available mathematical package that can produce bifurcation curves for equilibria as well as for periodic orbits, to get the complete bifurcation diagram of the system. Figure 8 shows the resulting bifurcation diagram where solid and dashed lines represent stable and unstable equilibria respectively while solid and dashed circles denote stable and unstable limit cycles. It should be noted that the amplitude of the limit cycles are much larger on the negative side as the ohmic effect of M_K was not considered in the model as it was not found necessary for the bifurcation. It can be seen that the limit cycles are born initially through a fold bifurcation of cycles. Both the Hopf bifurcations are subcritical as they involve an unstable limit cycle. The amplitude of the stable limit cycle keeps on reducing until it coalesces with the unstable limit cycle in another fold bifurcation. For the ramp experiment in SPICE, the second

bifurcation we see is actually the fold bifurcation of the limit-cycle. The Hopf bifurcation of the equilibrium occurred earlier but was not visible as the solution was in the basin of attraction of the limit-cycle. The unstable limit cycle acts as the threshold or basin boundary between the basins of attraction. In fact, this might be an alternative dynamical mechanism for the excitation block because in experiments, the subcritical Hopf bifurcation would not be visible since the basin of attraction of the equilibrium is much smaller than that of the limit cycle. So the fold bifurcation of the decreasing-amplitude limit cycle appears as supercritical Hopf bifurcation. For a comparison with HH model, its bifurcation picture is also shown in figure 9. The origin is made the equilibrium by a change of variable. It can be seen that the limit cycle is born by a fold bifurcation but terminates in a supercritical Hopf bifurcation. But the qualitative nature of the plots are similar as the reduction in amplitude of the limit cycle before its disappearance appears in both pictures.

Another interesting phenomenon because of the subcritical Hopf bifurcation is the presence of hysteresis-that is the neuron does not stop spiking even if I_{in} is reduced below the critical bifurcation value as its state is in the basin of attraction of the limit cycle. Thus, for a range of parameters, we have co-existence of two stable attractors, a limit cycle and an equilibrium. The unstable limit cycle acts as the threshold or basin boundary between their basins of attraction. The presence of the stable limit cycle is also observable in SPICE simulations. We conduct a simulation where I_{in} is a brief pulse of current. This pulse approximates a delta function, whose effect is to change the initial condition of the system by the area under its curve. Thus, the net amount of charge in the current pulse produces a change in voltage, ΔV_{mem} on C_{mem}.

Fig. 10 shows the effect of different size current impulses on the neuron. Smaller current pulses do not push the initial condition beyond the basin of attraction of the stable equilibria, but a large pulse shifts the initial condition beyond the unstable limit cycle leading to spontaneous oscillations.

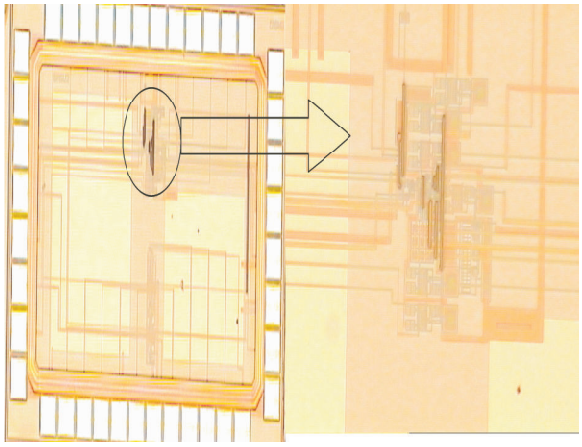


Figure 11: **Die photo:** A 1.5mm×3mm test chip fabricated in 0.35 μm CMOS. The chip has one sodium and one potassium channel and peripheral instrumentation to read out signals.

2.3 Hardware Platform

A test chip having one sodium and one potassium channels was used to test the bifurcation structure of the system. Figure 11 shows the photograph of the fabricated IC in TSMC 0.35 μm CMOS process. In the chip, the input current stimulus was created by varying the gate voltage of a PMOS transistor connected to V_{mem}. In this chip, the parameters were set using off-chip DACs. The output voltages were buffered out of the chip using voltage buffers. Also, there is a provision for configuring the channels for measuring voltage clamps.

An important aspect of the design was the ability to easily control the parameters of the equations independent of each other. Independent control is not enabled in the standard circuit shown earlier in Fig. 1. Fig. 12 shows a partial schematic of the chip to illustrate the solution to this issue. In the figure, transistors Mj' are of the same dimensions as Mj. If I_{amp} is changed, the equilibrium gate voltage of M1 changes and hence I_{tauh} changes too. A similar problem occurs for I_{taun} when V_{gk} is modified to change I_{K0}. To circumvent this problem, transistors M1' and M2' are used to recreate the bias voltage at the gate of M1. This voltage is buffered onto the source of M3' which now produces a copy of $I_{\tau h}$. Thus, $V_{\tau h}$ can now be changed to set the

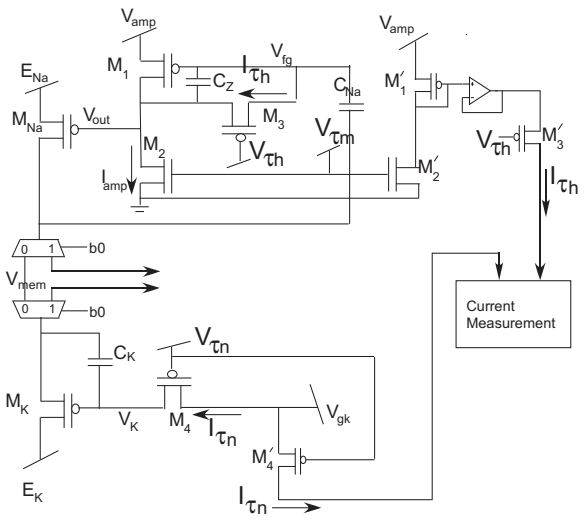


Figure 12: **Chip schematic:** Scheme used for independently controlling the parameters of the circuit. The transistors M1', M2' and the buffer recreate the source voltage of M3 at the source of M3' allowing measurement of $I_{\tau h}$. A similar argument holds for $I_{\tau n}$. The multiplexors are used to configure the chip for voltage clamps or current clamps.

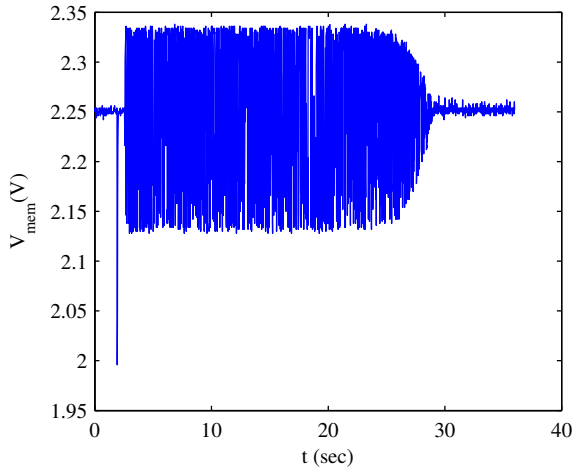


Figure 13: **Measured ramp experiment:** Measured data from an experiment where the input current is slowly increased in a linear fashion. The negative spike marks the beginning of the ramp where the current was reset after the last ramp. After a certain critical current the neuron starts oscillating. At higher currents the amplitude of the limit cycle reduces and finally the equilibrium becomes stable. This is very similar to Fig. 6.

desired value of $I_{\tau h}$ even if I_{amp} or V_{amp} are varied. A similar case arises for $I_{\tau n}$. The multiplexors allow the circuit to be configured for voltage clamp (b0 is set high) or current clamp (b0 is set low). When configured in the voltage clamp configuration, the parameters I_{Na0} and I_{K0} can be measured and set appropriately.

Figure 13 shows measured data from this chip. The bifurcations were observed by ramping the input current up in a linear fashion by sweeping the gate of the PMOS in a logarithmic fashion. The initial negative spike marks the start of the ramp where the current was reset from the end of the previous ramp. As expected, we see two bifurcations by which the neuron starts and stops oscillating. As expected, the first bifurcation is a subcritical Hopf bifurcation. The second bifurcation looks like a supercritical Hopf bifurcation because noise caused the system to oscillate even though the equilibrium might have become stable, which was discussed earlier. This looks slightly different from Fig. 6 as the initial and final currents and the rate of increase of current were different in measurements and simulations. However, the important point is the qualitative similarity in the two figures.

2.4 Conclusion

In this chapter, we modeled a silicon neuron and demonstrated that it exhibits Hodgkin-Huxley type dynamics. There have been multiple instances of modeling Hodgkin-Huxley equations in silicon. Almost all the implementations use the Hodgkin-Huxley (H-H) formalism, some of them using a set of equations simpler than the original H-H equations [18] [77] while others have modeled in detail the full set of equations [69] [95] [2] [40] [96]. However, the principle of this design is based on similarities between voltage clamp experiments on transistor channels and biological ion channels [37]. We extend the work in [37] that showed single action potentials and present a detailed non-linear dynamic analysis of the neuron circuit for different bias regimes. In particular, we show a subcritical Hopf bifurcation which is the trademark of Class 2 neural excitability (observed for example in brainstem mesencephalic V neuron [55]). Also, we demonstrate a bifurcation mechanism involving subcritical Hopf bifurcation and a fold limit cycle bifurcation that models the excitation block

phenomenon observed in many neurons such as layer 5 pyramidal neuron of rat's visual cortex [55]. Hence this chapter strongly validates the method of using transistors to model channels by showing the qualitative similarity in the dynamical behavior of this circuit with biology across parameter ranges of interest.

This work enables finding the proper biasing regime for this circuit, a non-trivial task because of the large dimensionality of the parameter space. Also, this is the first low-power on-chip implementation of the circuit as [37] mentions using large bias currents and off-chip capacitors. Now the power of this compact implementation can be exploited by integrating multiple such channels on a chip together with synapses and dendrites. Next, a neuron design that exhibits saddle-node bifurcation will be discussed.

CHAPTER III

A SADDLE-NODE NEURON AND ITS NULLCLINE BASED DESIGN

A classification of neural spiking was proposed by Alan Hodgkin [48] where he classified neurons in three broad categories. The first two were capable of repetitive spiking. Type I neurons possessed F-I (current-frequency) curves that approached zero, i.e. the neurons were capable of firing at large as well as arbitrarily low frequencies, while type II neurons transitioned from silence to firing at an arbitrary non-zero frequency. Subsequent analysis, put forward in [87] and summarized in [55] has shown that these two mechanisms ubiquitously describe nearly all spiking neuron models. Furthermore, the type I characteristic is uniquely associated with a saddle-node bifurcation at the transition from silence to spiking, and the oscillation (spiking) responds to depolarizing perturbations with a phase-advance. The type II characteristic is uniquely associated with a Hopf bifurcation, and the oscillation (spiking) responds to a depolarizing perturbation by either advancing or delaying the oscillation, depending on where in the limit cycle the input arrived.

All the current silicon designs typically use a neuron exhibiting class I behavior. Most of these designs have used the integrate and fire (I&F) model and its variants because of its simplicity. These phenomenological models however fail to capture many properties of actual conductance-based neurons like phase response curves. On the other hand, some designs have faithfully replicated full Hodgkin-Huxley (H-H) dynamics [89, 95, 115] resulting in large footprints for each neuron.

Our silicon neuron with class I membrane dynamics lies in a space between these

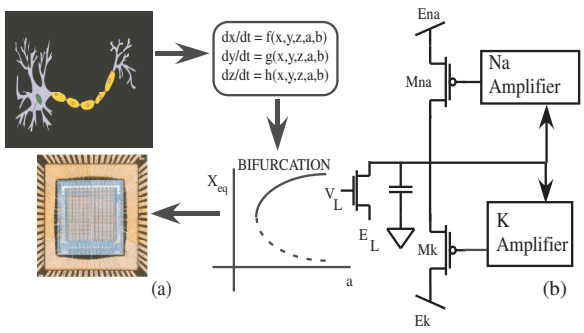


Figure 14: **Silicon neuron design:** (a) The methodology for creating silicon neurons that we use involves designing for certain desired bifurcations. (b) Our concept of silicon neurons has transistors modeling biological channels (sodium, potassium etc.) with amplifier circuits sensing the membrane voltage and changing the conductance of the channel transistors appropriately.

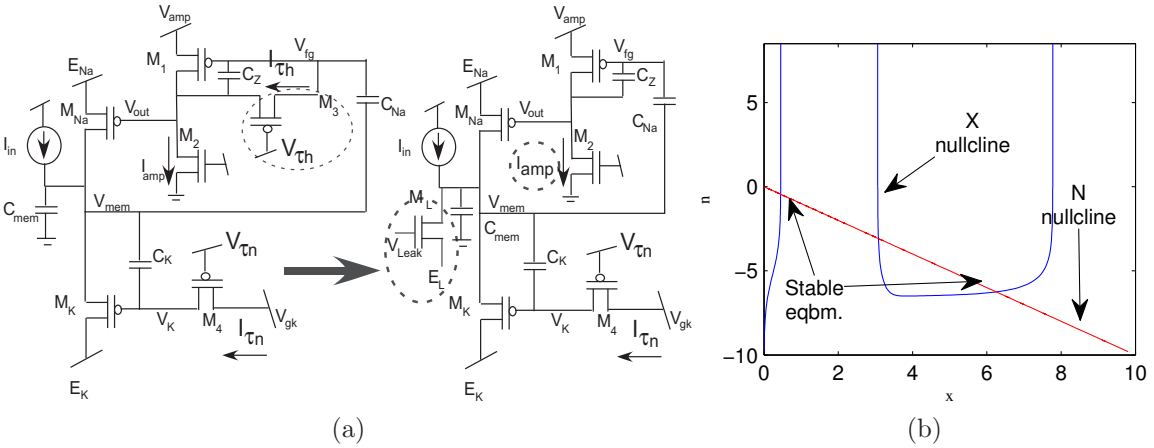


Figure 15: **Transformation from Hopf neuron:**(a) Removing the transistor responsible for inactivation of the sodium current results in a persistent sodium current which can produce multiple equilibria. A leak channel is added to create the cubic nullcline. I_{amp} is increased to allow for approximating sodium activation as instantaneous. The modified elements are encircled. (b) Nullclines of the new neuron structure shows three equilibria, two of which are stable and one is unstable. The equilibrium at high membrane potentials is not found in biology. Here, x and n signify normalized membrane potential and potassium activation respectively.

two. Figure 14(a) demonstrates the basic philosophy: the differential equations modeling ion-channel dynamics in a neuron exhibit certain bifurcations which are responsible for its computational properties [55]. We make circuits that exhibit similar bifurcations but do not necessarily model the original differential equations. This allows us to create more compact and power-efficient designs. Figure 14(b) shows a schematic of the neuron where a transistor is used to model the biological channel

while a separate circuit senses the membrane voltage and applies the appropriate voltage on the gate of the channel transistor to modulate its conductance. The feedback circuit can be designed based on the desired bifurcation sequence (which implies a certain set of associated phenomenon like positive feedback, phase response curve [55]) or on the voltage clamp responses of individual channel modules. The earlier chapter and associated paper [13] demonstrated class II excitability using a similar approach. Here, we derive the circuit for class I membrane dynamics starting from the earlier design in [13, 37] and also develop methods for algorithmically biasing the circuit in the correct regime based on nullclines (curves along which the vector field is zero horizontally or vertically) of the reduced two-dimensional model. The circuit we present for extracting the nullclines coupled with a method for setting local biases (e.g. floating-gates) in a neuron array shall also allow us to reduce mismatch induced variations in the neuron array. We describe these features in the following sections.

3.1 Circuit Operation and Model

In this section, we start from the Hopf neuron model presented earlier [13] and follow a route of reducing this model to one that can exhibit saddle-node bifurcations (we call it a 'saddle-node neuron'). We deduce theoretically that changes have to be made to the K-channel structure to get the desired behavior. The existence of the oscillations can be understood from studying the nullclines for the model.

3.1.1 Hopf neuron and its modification

The circuit on the left in Fig. 15(a) shows the Hopf neuron that we have presented earlier. We consider these circuits to be composed of individual channels. Each channel has a channel transistor that supplies the current and a gating amplifier that controls the gate of this transistor in some non-linear way to produce desired dynamics. This circuit has only one equilibrium [13] as can be seen from its monotonic I-V curve (the number of zero crossings of the I-V curve show the number of equilibria). This is

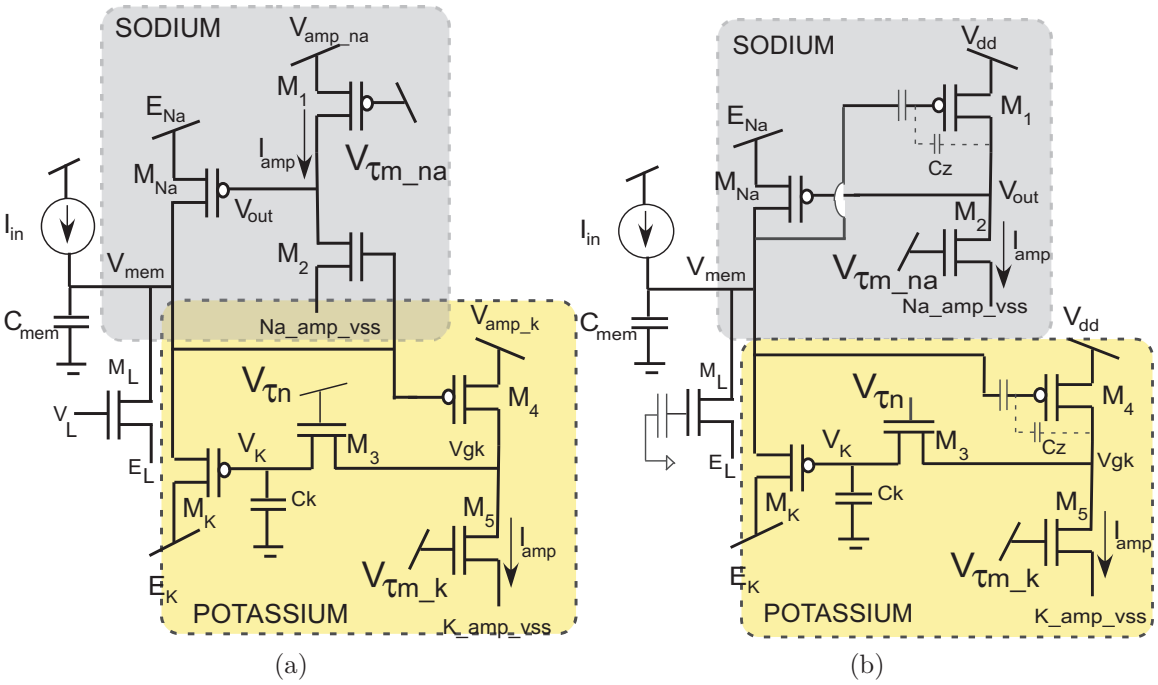


Figure 16: **Saddle neuron:**A compact silicon neuron that exhibits saddle-node bifurcations of the equilibrium. The limit cycle might be born by SNIC or a saddle-homoclinic loop bifurcation. (a) and (b) show the non floating-gate and floating-gate (FG) versions respectively. To ensure that the n-nullcline can intersect the x-nullcline in the twice in the middle branch, the gain of the potassium amplifier has to be larger than the sodium amplifier. Spike frequency adaptation can be obtained by adding another similar potassium channel with a slower activation time constant set by M3.

because the gating amplifier for the transient sodium channel is bandpass leading to the sodium channel transistor acting as approximately a constant current source in steady state. Hence the I-V relation for this neuron is dictated by the monotonic, exponential I-V relation of the potassium channel transistor.

3.1.1.1 Absence of inactivation

In order to create a saddle-node neuron there must be multiple equilibria, one of which should be of the saddle type. Thus the neuron must have a non-monotonic I-V curve with multiple zero-crossings. To create a non-monotonic curve, we must have a positive feedback channel that creates a negative conductance region (since there has to be a part of the I-V curve where current increases even though voltage across the channel decreases). This implies that the positive feedback sodium channel in

this implementation needs to be a low-pass one instead of a bandpass one so that it can affect the DC/persistent current of the channel (this model is typically called 'persistent sodium' model as compared to the HH model which is a 'transient sodium' model). Hence, the feedback transistor providing inactivation current $I_{\tau h}$ needs to be removed as shown in the right half of Fig. 15(a). The fact that the low-pass amplifier does indeed give saddle-node bifurcation can also be seen from the equations in [13] by a change of variables and letting $I_{\tau h} \rightarrow 0$. Essentially, a simple common source amplifier or any other DC-coupled amplifier may also be used as the low-pass amplifier.

3.1.1.2 Instantaneous Activation

The typical way to simplify the model further [55] is to assume that the sodium activation is much faster than the potassium; so it can be assumed instantaneous. This amounts to assuming $I_{amp} >> I_x$ where I_x is any other bias current in the circuit. Thus, we have a 2-dimensional system with the membrane potential and the voltage at the gate of the potassium channel transistor being the only two variables. The other major difference from the Hopf neuron is the need for a leak channel as shown in Fig. 15(a). This is needed to create the negative part of the I-V relation to ensure zero crossings. Assuming a hyperbolic-tangent (tanh) function to approximate the amplifier's characteristic, applying Kirchoff's current law (KCL) to the neuron in a way similar to [13] yields:

$$\begin{aligned}
\dot{x} &= e^{-m_{ss}(x)}(1 - e^{x-8}) + i - I_{k0}e^{-n-x}(e^x - 1) \\
&\quad - I_{l0}(1 - e^{El-x}) \\
\dot{n} &= r(1 - e^{n+x}) \\
m_{ss}(x) &= -\alpha_1 \tanh(\frac{x - o_1}{s_1})
\end{aligned} \tag{12}$$

where $x = \delta V_{mem}/U_T, n = \delta V_k/U_T, r = I_{\tau n}/C_k, t' = t/\tau, \tau = C_{mem}U_T/I_{na0}$ and the derivatives are taken with respect to t'. All the currents in the equation are

normalized by I_{na0} while capacitors are normalized by C_{mem}. Also, parameters 'i', 'o_1' and 's_1' represent the input current stimulus, the amplifier's trip-point and gain respectively. The phase portrait for this system is drawn in Fig. 15(b). The coupling of a transistor's gate to the surface potential, κ, has also been assumed to be 1. The variables x and n signify normalized membrane voltage and potassium channel activation respectively.

3.1.2 Motivation for changing the Potassium channel amplifier

It can be seen from Fig. 15(b) that the x-nullcline now has three branches (a typical 'N' shape). So there is a possibility of multiple intersections with the other nullcline and hence the existence of multiple equilibria. However, the equation for the n-nullcline in (12) is linear and can be written as n=-x. Hence, the n-nullcline will intersect the x-nullcline in all three branches. However, it is well-known [55] that most of the middle branch of the x-nullcline in this type of a system corresponds to unstable equilibria, while the other two branches indicate stable equilibria. So, in this case we will have a stable equilibria at a high value of membrane potential for a large range of parameter values, something not commonly observed in biology. The existence of this equilibria might also be understood intuitively. At low membrane potential values, the sodium current will be balanced by a leak current and the potassium current. As the membrane potential increases, the sodium current increases a lot but then starts to decrease when the sodium-channel amplifier saturates at Na_amp_vss. At the same time, the potassium channel current is increasing and hence, there should exist a high value for V_{mem} where they are equal and perturbations to that state also die down.

We desire to have a saddle-node on invariant circle (SNIC) or a saddle-homoclinic loop (SHM) bifurcation with increasing input current. In most neural models, the limit cycle born from these bifurcations is the only stable attractor [53] [87]. This

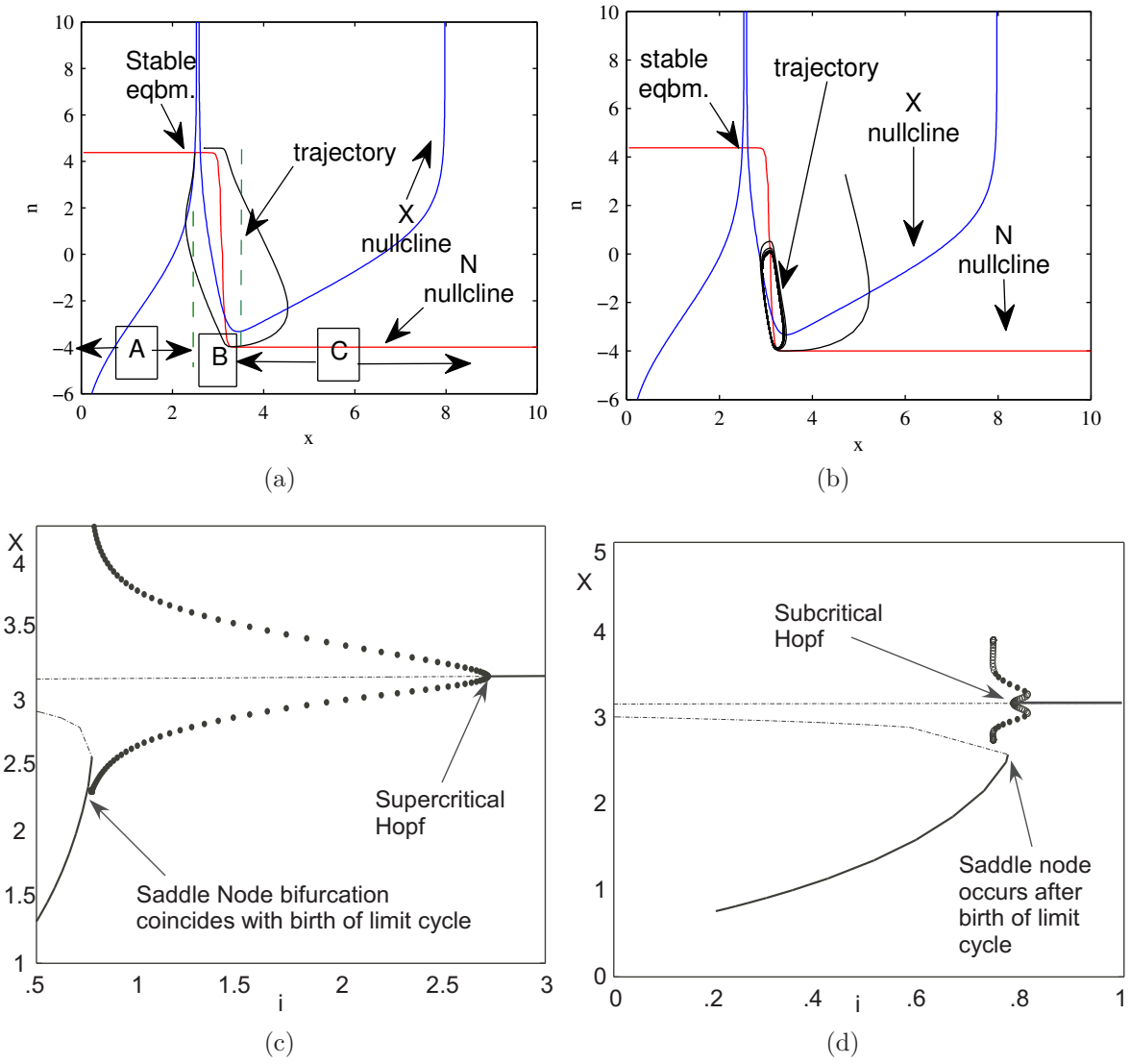


Figure 17: **Phase portrait and Bifurcation diagram**(a) The phase plane for the system with slow activation of K-channel. The x-nullcline has three distinct regions. In region A, x changes from lower than E_l to larger than it resulting in leak current overpowering sodium current. In region B, x is close to o_1, while in region C, the sodium amplifier has saturated at Na_amp_vss. It can be seen that the trajectory starting from the right of the saddle makes a big loop before returning to the stable equilibrium. (b) The phase plane for fast potassium activation leading to saddle-homoclinic loop bifurcation. Even before the saddle-node bifurcation, a limit cycle has been created by a homoclinic bifurcation from the saddle equilibrium. (c,d) Bifurcation diagram for the case in (a,b)

implies that there should not be any other stable equilibria when the saddle-node bifurcation which involves the equilibrium at low membrane potential occurs. So the n-nullcline has to intersect the x-nullcline once in the lowest membrane potential branch and twice in the middle branch implying that it must have a sharply falling

part. The sharply falling part of the nullcline can be created needs to be an inverting amplifier that drives the gate of M_k with the desired characteristic. This leads to a circuit for the saddle-neuron as shown in Fig. 16 where both the floating-gate (FG) and non-FG versions are shown. The FG version has the advantage of requiring two less biases and can be more compact without trading-off transistor threshold voltage matching. It will be shown in section 3.3 that using only three FG transistors as depicted in Fig. 16(b) is sufficient to eliminate the sources of mismatch in the circuit. In this circuit, M1 and M2 form the Na-channel inverting low-pass amplifier while M4 and M5 form the K-channel inverting low-pass amplifier. In the FG version, capacitors C_z (which can be explicitly drawn capacitors or overlap capacitors associated with the transistors) set the gain of the gating amplifiers. To ensure that the n-nullcline can intersect the x-nullcline twice in the middle branch, the gain of the potassium amplifier has to be larger than the sodium amplifier. I_{amp} is chosen large enough so that the activation kinetics of both the amplifiers are faster than other time scales in the circuit. The slow K-channel time scale is controlled by M3. We could have used a PMOS or a resistor in its place which would result in slightly different shapes of the spikes. The equations to model this circuit are as follows:

$$\begin{aligned}
\dot{x} &= e^{-m_{ss}(x)}(1 - e^{x-8}) + i - I_{k0}e^{-n}(e^{x} - 1) \\
&\quad - I_{l0}(1 - e^{El-x}) \\
\dot{n} &= r(e^{-n} - e^{-n_{ss}(x)}) \\
m_{ss}(x) &= -\alpha_1 \tanh(\frac{x - o_1}{s_1}) \\
n_{ss}(x) &= -\alpha_2 \tanh(\frac{x - o_2}{s_2})
\end{aligned} \tag{13}$$

with similar interpretation of variables as in equation (12). Parameters o_1 and o_2 can be modified by changing V_{amp_na} and V_{amp_k} in the non-FG version or by changing the charge on the gates of M1 and M4 respectively. $\alpha 1$ and $\alpha 2$ are set by the difference in the power supply voltages of the amplifiers. Spike frequency adaptation can be

Table 1: Parameters for Numerical Simulation

Parameter	α_1	α_2	o_1	o_2	s_1	s_2
Value	2.3	4	3	3.1	0.25	0.06
Parameter	I_{l0}	I_{k0}	E_l	r(SNIC)	r(SHM)	i
Value	1	0.01	0.4	0.7	1.3	0-3

obtained in this neuron by adding another similar potassium channel but with slower activation kinetics compared to the potassium channel responsible for regular spiking.

It should be noted that (13) is similar to the traditional Morris-Lecar model [63] which is obtained following a similar reduction principle from the Hodgkin-Huxley model. The equations for this model are:

$$\begin{aligned} C\dot{V} &= g_{Ca} m_{ss}(V)(E_{Ca} - V) - g_K W(V - E_K) \\ &\quad - g_L(V - E_L) + i \\ \dot{W} &= \lambda(V)(w_{ss}(V) - W) \\ m_{ss}(V) &= \frac{1}{2}\{1 + \tanh\left[\frac{V - V_1}{V_2}\right]\} \\ w_{ss}(V) &= \frac{1}{2}\{1 + \tanh\left[\frac{V - V_3}{V_4}\right]\} \\ \lambda(V) &= \lambda_0 \cosh\left[\frac{V - V_3}{2V_4}\right] \end{aligned} \tag{14}$$

The differences between (13) and (14) stem from using transistors with exponential characteristics in place of channels with linear characteristics.

3.1.3 Phase Portrait and Bifurcation Diagrams

In this subsection, the numerical simulations of (13) are described. Bifurcation diagrams for varying input current are computed using the continuation software AUTO [35]. Two different cases are considered based on the speed of the potassium activation. The parameters for the numerical simulation results shown in Fig. 17 are presented in table 1.

Figure 17(a) depicts the phase portrait of the system with a relatively slow potassium activation time constant (smaller $V_{\tau n}$) and a fixed value of input current that is close to the bifurcation value. The nullclines of the system shown in the figure do not change with the activation time constant. There are three intersections of the two nullclines, as desired, leading to two unstable and one stable equilibrium. The middle equilibrium is a saddle-node and is about to collide with the stable equilibrium to create a bifurcation. It can be seen that the trajectory starting from the right of the saddle makes a big loop before returning to the stable equilibrium. At the saddle-node bifurcation, this trajectory becomes homoclinic to the saddle, and after the bifurcation forms a stable limit cycle. As the homoclinic loop is an invariant manifold homeomorphic to a circle, this is also called saddle-node on invariant circle (SNIC).

From (13) it can be seen that the n-nullcline is given by $n_{ss}(x)$ as shown in Fig. 17(a) or (b). To study the structure of the x-nullcline, one must understand that the n-axis corresponds to the voltage on the gate of M_k required to make $\dot{x} = 0$, i.e. to balance the rest of the current incident on the membrane capacitor. The nullcline can be divided into three distinct zones marked A, B and C in Fig. 17(a). For $x < E_l$, there exists a value of n to balance the leak current and the sodium current. But for x sufficiently larger than E_l such that the leak current is larger than the sodium current, no value of n exists since the polarity of the leak current is then similar to the potassium current. So the x-nullcline shoots up to infinity near the end of zone A. In zone B, x is close to o_1 and the sodium control amplifier is in the high gain region. The slope of the nullcline in this region is $\frac{1-A\kappa}{\kappa}$ where A is the gain of the sodium amplifier and κ is the coupling strength of the gate to surface potential for a transistor. Lastly, in zone C, the sodium amplifier has saturated at Na_amp_vss implying that n changes to compensate for the change in potassium current due to the change in membrane potential. So the slope in this region is $1/\kappa$. The nullcline

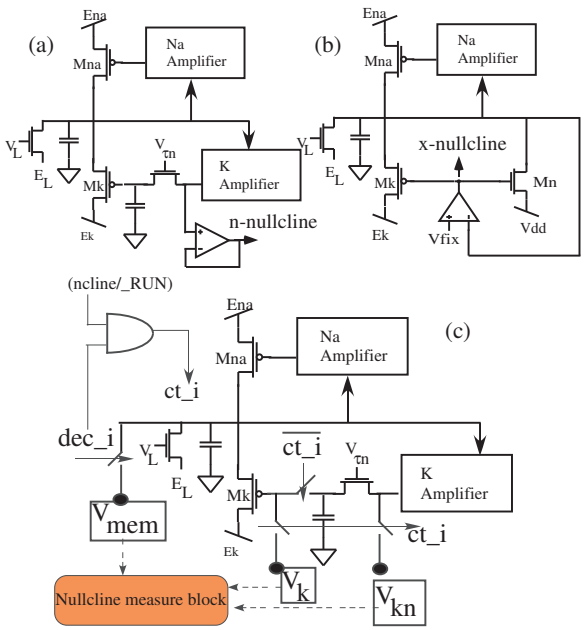


Figure 18: **Circuits to extract the nullcline:** (a) The n-nullcline for the system is a curve n_1(x) such that $\frac{dn}{dt}$=0 on this curve. This is same as the characteristic of the K-amplifier. (b) The x-nullcline is a curve n_2(x) such that $\frac{dx}{dt}$=0 on this curve. Finding the value of n on this curve corresponding to a certain value of $x = V_{fix}$ (say) is equivalent to forcing the desired x using an amplifier connected in feedback controlling the gate of M_k. The desired n value is created at the output of the amplifier. An additional NMOS (M_n) is used for bi-directional capability to avoid getting attracted to an undesired fixed point. (c) Modification for integrating these neurons into an array with the selection, biasing and nullcline measurement circuitry at the periphery.

changes sharply at the ends when M_{na} or M_k are in the ohmic region.

Figure 17(c) depicts the bifurcation diagram in this case which shows the birth of a limit cycle coinciding with a saddle-node bifurcation. The limit cycle ends in a supercritical Hopf bifurcation for larger values of input current.

Figure 17(b) shows the phase portrait for the case where the activation of potassium is relatively fast. As can be seen from the figures 17(b) and (d), even before the saddle-node bifurcation, a limit cycle has been created by a homoclinic bifurcation from the saddle equilibrium. The main observable difference from SNIC is the bistability of the system before the saddle-node bifurcation in a certain parameter range. Intuitively, this is because the potassium current deactivates quickly and does not allow the membrane potential to undershoot the stable equilibrium voltage. The limit cycle, in this case, ends in a subcritical Hopf bifurcation at a larger current

value.

3.2 Nullcline Extraction and Algorithmic Biasing

The last section described the role of nullclines in determining the dynamics of this circuit. It is of utmost importance to be able to extract the nullclines of the fabricated system and then vary the parameters to get the right shape and intersections. In this section, we propose a circuit to do so and also describe an algorithm for biasing the circuit based on the measured nullclines.

3.2.1 Nullcline Measurement circuit

Figure 18(a) and (b) show methods to measure the n and x-nullcline of this system. Note from eq. (12) that x and n denote normalized values of the membrane voltage and potassium channel gating voltage respectively. As mentioned earlier, the n-nullcline of the system is given by $n_1(x) = n_{ss}(x)$ since that is the value of n for which the current charging C_k is 0. This is exactly equal to the characteristic of the potassium gating amplifier. Hence, we only need to monitor the output voltage of the potassium amplifier to get this nullcline.

The x-nullcline is a curve n_2(x) such that $\frac{dx}{dt}$=0 on this curve. Finding the value of n on this curve corresponding to a certain value of $x = V_{fix}$ (say) *is equivalent to forcing the desired x using an amplifier connected in feedback controlling the gate of M_k.* The desired 'n' value is created at the output of the amplifier. This structure is similar to a logarithmic amplifier that we have used earlier for measuring low currents [14].

However, one can see from Fig. 18(b) that there is an extra NMOS, M_n also connected in feedback across the amplifier together with M_k. This structure is actually useful in measuring bi-directional currents [12]. One may ask 'Why is it needed ?'. This is necessary because otherwise the circuit may be attracted to an undesired fixed point. We can understand the existence of that fixed point if we consider the

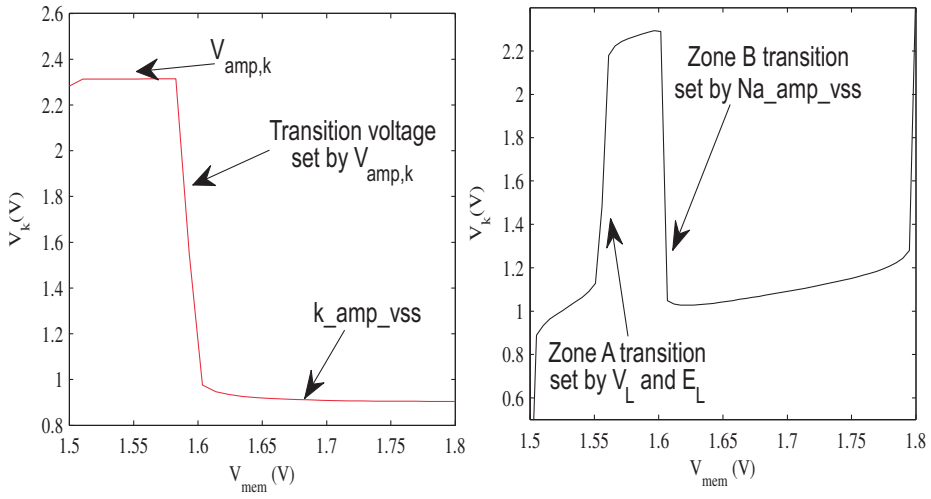


Figure 19: **Control of Nullcline:** Different regions of the nullcline can be controlled by the biases in the circuit.

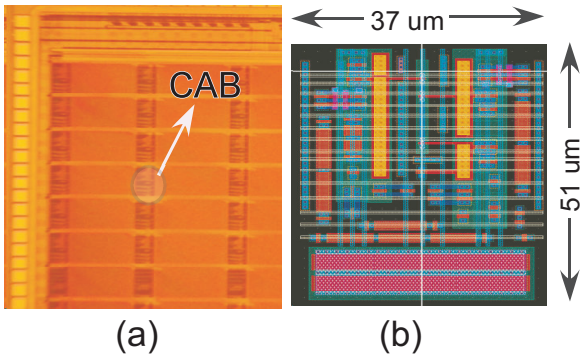


Figure 20: **Die Photo and Layout:** (a) Die photo of the fabricated FPAA in 0.35 μm CMOS showing a CAB with low-pass sodium and potassium channels. (b) Layout of the optimized FG based neuron cell for a custom chip in the same process.

following thought experiment: consider increasing the value of the membrane voltage from E_k to E_na. At values of V_{fix} slightly larger than the first equilibrium, the current through M_k is very small since the sodium current is balanced by the leak and so the amplifier tries to shut off M_k by pulling its gate high. Beyond this point, the leak current is larger than the sodium current; hence the current through M_k needs to be negative to satisfy $dx/dt = 0$, but the best the amplifier can do is to push the gate of M_k to V_{dd}. This results in the feedback loop being broken and the membrane is no longer clamped to V_{fix}. Instead it goes to equilibrium at the earlier value based on just the leak and the sodium channel. This value is lesser than the present value

of V_{fix} which means increasing V_{fix} more only keeps the amplifier railed to V_{dd} and V_{mem} at its earlier equilibrium. Adding the NMOS allows the amplifier to clamp the membrane while sinking or sourcing currents. It forces the membrane to V_{fix} always and does not allow the feedback loop to be broken.

Having introduced the nullcline extraction circuits, a natural question to explore is the area penalty for including this circuit in an integrated circuit. Since it is desirable to maximize the number of neurons in a chip, requiring an amplifier in every neuron would be prohibitive. Fortunately, it is possible to share one nullcline measuring circuit for an entire array of neurons. Figure 18(c) depicts the modifications needed to enable this for the 'i-th' neuron in an array. A control signal, $ncline/\_RUN$ puts the neuron array in nullcline measuring or normal operational mode. Apart from the common biasing of the neurons, three other lines are shared among the neurons: V_{mem}, V_k and V_{kn}. Switches controlled by a decoder are introduced in every neuron so that the membrane potential, the gate of M_k and the output of the potassium gating amplifier are connected to those three global lines in the nullcline measuring mode. The amplifier and other selection circuits can now be confined to the nullcline measurement block at the periphery of the array.

3.2.2 Algorithmic Biasing

Using the circuits described earlier, we can extract the nullcline of the circuit. Now, we describe a methodical approach to setting the biases for the non-FG circuit based on the measured nullclines with a goal of obtaining a saddle-node bifurcation with increasing current. The floating-gate based circuit can be biased in a similar way.

1. In the beginning, V_{amp_na}, V_{amp_k}, Na_amp_vss, K_amp_vss and $V_{\tau n}$ are set to V_{dd} to shut off the gating amplifiers and make M3 act as an ON switch. V_L is set to zero to turn off the leak channel too. The difference between E_{na} and E_k is a system level specification, typical values being 200-300 mV. The absolute value of these bias

voltages are kept close to the middle of the chip's power supply to allow for sufficient headroom in setting the gating amplifiers' power supplies.

2. Set E_l to the value of V_{mem} where it is desired for zone A of the x-nullcline to end.

3. The value of V_{amp_na} is initially set at V_{dd}. $V_{\tau m\_na}$ is set so that I_{amp} is large enough to set a time constant much faster than that of the slow potassium (typical values are 0.5-1 nA). A good strategy is to set I_{amp} much higher initially and then reduce I_{amp} till the spike shape changes.

4. Na_amp_vss is then reduced gradually from V_{dd}. This shifts zone B of the x-nullcline to progressively lower voltages. Continue till zone B is in the desired range of membrane potentials.

5. Increase V_l till the x-nullcline has the characteristic of zone A and increase sharply to a high voltage. Increase V_l by approximately 10-20 mV to ensure a high enough threshold current. Since V_l sets the threshold current (exponential relation), its final value can be set after measuring the graph of its dependence.

6. K_amp_vss is reduced to approximately 50-100 mV below the lowest value of the measured x-nullcline. $V_{\tau m\_k}$ is now set using a strategy similar to the setting of $V_{\tau m\_na}$ for a particular I_{amp}.

7. V_{amp_k} is slowly reduced from V_{dd}. This shifts the n-nullcline to lower membrane potentials. Continue till the n-nullcline crosses the x-nullcline in zone B.

To verify that the circuit does have a limit cycle, it can be configured as neuron and input current applied to check for oscillations. How the different biases affect parts of the nullcline is shown in Fig. 19.

Table 2: Parameters for the Fabricated Circuit

Circuit Element	Value	Bias	Voltage
M_{Na}	5/1	E_k	1.5
M1	1/.4	E_{na}	1.8
M2	10(10/.4)	V_{amp_na}	1.7
M_k	10/1	$V_{\tau m\_na}$	1.17
M4	5(10/6)	Na_amp_vss	1.24
M5	2(10/6)	E_l	1.55
M3	2(10/6)	V_l	2.25
C_k	300 fF	V_{amp_k}	2.01
Ml	5/2	$V_{\tau m\_k}$	1.24
		K_amp_vss	0.8
		$V_{\tau n}$	1.55

3.3 Measurements

The proposed neuron and related concepts have been verified by fabricating an IC with the neuron circuit in 0.35 μm CMOS process and is shown in Fig. 20(a). The fabricated IC has an architecture similar to a field programmable analog array (FPAA) as described in [15] and also detailed in the later chapters of this thesis. The architecture consists of an array of computational analog blocks (CABs) embedded in a sea of switches that can be configured to be ON or OFF as desired. The elements in the CABs included operational transconductance amplifiers, synapses and different types of channels. The switches needed for configuring the neuron into operational mode or nullcline measurement mode as described in section 3.2 were not explicitly needed since the reconfigurable switch matrix could perform that task. Since the neuron was placed in a CAB, it was not optimized for size and included individual biasing circuits. The resulting area is 2740 μm^2 including the extra potassium channel for spike frequency adaptation. The poly-poly capacitor took up a major share of the area, an aspect that can be largely improved by using a moscap instead. The transistor sizes

and bias voltages are tabulated in 2. Note that the channel length of transistors in the potassium gating amplifier were made much larger than the sodium amplifier to make it have a larger gain as noted in section 3.1. An optimized layout of the neuron including an extra potassium channel for spike frequency adaptation has now been done for a custom chip as shown in Fig. 20(b). This FG based neuron occupies a much smaller area of 37×51 μm^2.

3.3.1 Nullclines, SNIC and SHM bifurcations

Figure 21(a) shows measured n-nullclines from the fabricated chip. As expected, reducing V_{amp_k} reduces the trip point of the potassium amplifier (or o_2) resulting in the high-gain part of the activation curve moving to lower membrane potentials. Figure 21(d) depicts the measured x-nullcline for different values of input current. As mentioned earlier, the sharply rising part at the end of zone A in Fig. 17(a) is because of the leak current overpowering the sodium current. However, an increase in the input current can balance the leak as is seen in the figure. This confirms the existence of the saddle-node when the channels are combined together.

Figure 21(b) and (e) are measured spiking waveforms from the neuron. Figure 21(b) corresponds to a lower value of the input current that is slightly larger than the bifurcation value. The variability in the inter-spike interval is due to ambient noise. The shape of the spikes are typical of SNIC bifurcations and type-I membrane dynamics. In the period right after a spike, there is a hyperpolarization of the membrane followed by a long time when the membrane appears to be at equilibrium. In this time, the solution is passing through the region of the phase space where the saddle-node existed earlier. The resulting small vector fields lead to the sluggishness of the trajectory.

Figure 21(c) shows the relation between the threshold current (I_{th}) for repetitive spiking and V_l or the parameter setting the leak conductance. The relation is

approximately exponential suggesting that the bifurcation occurs almost when the input current balances the leak current. Figure 21(f) shows the measured current-frequency behavior (f-i curve) of the neuron for a certain value of I_{th}. It exhibits a typical sub-linear behavior (theoretically square root in the neighborhood of the bifurcation).

The existence of the saddle-homoclinic bifurcation leading to bistability was also explored. Keeping the input current at a value slightly below the bifurcation value, $V_{\tau n}$ was slowly increased to speed up the potassium activation kinetics. Though it was not possible to obtain a parameter value where the equilibrium was stable and a perturbation could carry it into the basin of attraction of the limit cycle, intermittent spiking was observed for certain ranges of $V_{\tau n}$. The resulting waveform is shown in Fig. 22. We believe that the saddle-homoclinic bifurcation occurs in the circuit for this range of parameter values but the ambient noise carries the solution back and forth between the basins of attraction of the equilibrium and the limit cycle. A similar phenomenon is observed in the mathematical model when Gaussian noise is added to the input current. The susceptibility of the circuit to noise, random or otherwise (e.g. 60 Hz, digital switching of USB communication port on the board) increases as the bias currents and power supply values are reduced.

3.3.2 Spike Frequency Adaptation

A common feature of neurons that has been widely modeled is spike frequency adaptation. It was mentioned in section 3.1 that this phenomenon could be observed in this neuron model by adding a slower potassium channel along with the regular ones. This channel models a calcium gated potassium channel. The reconfigurability of the FPAA platform allowed us to verify this concept easily. The biases for this potassium amplifier were kept the same as the regular potassium amplifier. The activation time constant for this channel was increased by reducing $V_{\tau n2}$ by 250 mV relative to

the regular potassium channel. The input current to the neuron was stepped from a value lesser than the bifurcation value to one where the neuron spikes. The resulting waveform is plotted in Fig. 23. It clearly displays increasing inter-spike interval as it continues spiking. To decrease power consumption and chip area, the new design of the neuron shares the potassium amplifier for both the slow and fast potassium channels.

3.3.3 Power Dissipation

A major motivation in developing this design was to achieve the lowest possible power dissipation. For the neuron with spike frequency adaptation ,there are three gating amplifiers each of which was biased at a current of 0.5 nA. The other component of the power dissipation is the switching power which has been measured for several spike frequencies. The resulting plot is shown in Fig. 24. As expected, the switching power increases almost linearly with frequency. It should also be noted that the switching power is not the dominant factor in the net power dissipation. It may be possible to reduce the static power even further since the spike shape was not distorted till this point. With a power consumption of 1.74 nW at a frequency of 100 Hz, this neuron has the lowest power consumption among all reported designs to the best of the authors' knowledge. The new design of the neuron that shares the gating amplifier for both the potassium channels is expected to reduce power consumption further.

3.3.4 Mismatch and Variability

A major concern in the design of arrays of spiking silicon neurons is mismatch among the neurons. The two parameters, whose matching we are concerned with in this design, are the f-i curve and the bifurcation current value corresponding to the birth of the limit cycle. Since biases are shared across neurons in the array shown in Fig. 18(c), matching of the neuron parameters would require very large sized transistors increasing the footprint of one neuron. Instead, we plan to use floating-gate (FG)

Table 3: Performance comparison of silicon neurons

Reference	Model detail	Power @100Hz	Area	Process	FOM/1e3	Methodology
This work	Bifurcation based, SFA, PF, RP[1]	1.74 nW	2740 μm^2 [2] or 37×51 μm^2 [3]	0.35 μm	37.3	V-mode nullcline based design
[67]	Conductance based, SFA, PF, RP[1]	3.3 nW(sim)	913 μm^2	0.35 μm	40.7	I-mode, Differential pair integrator instead of log-domain
[38]	Mihalas-Niebur	7.5 nW(sim)	70×70 μm^2	0.15 μm	0.46	Switched Capacitor
[112]	Conductance based integrate and fire	64.5 nW	40×60 μm^2	0.5 μm	0.24	Switched Capacitor
[51]	Integrate and fire, SFA, PF, RP[1]	0.5 μW	83×31 μm^2	0.8 μm	0.124	Modified Integrate and fire
[115]	Full H-H	0.3 mW[1]	1.5×1.5 mm[2][6]	0.5 μm	3.7e-7	I-mode multiplier and V-mode differential pair
[95]	Full H-H	1 mW	0.26 mm^2	1.2 μm	5.5e-6	I-mode multiplier and V-mode differential pair
[89]	Full H-H	-	834×870 μm^2	0.35 μm	-	I-mode
[102, 103]	Bifurcation based	-	38 transistors	discrete	-	V-mode design followed by linear V-I
[5]	Leaky integrate and fire, PF, RP[1]; Conductance based	-	28×36 μm^2	0.25 μm	-	log-domain I-mode
Digital	Izhikevich model	$\approx$ 1 μW	-	TI TMS320 C5X	-	DSP design
Biology	Full H-H, Dendrite, Adaptation and more	$\approx$ 20 pW	-	-	-	voltage and ligand gated channels, chemical signalling

[1] SFA: Spike frequency adaptation; PF: Positive feedback; RP: Refractory period
[2] FPAA based design
[3] Custom chip

transistors as shown in Fig. 16(b) that decouple the constraint of transistor size dictating mismatch. Of course, a programming circuit is now needed. But since we can share one such circuit for the entire chip, it is not prohibitive.

A naive solution would be to replace all the transistors in the circuit with floating-gates; but this would result in a severe area penalty because of the added selection circuitry per FG transistor in each cell. Instead, an analysis of this neuron circuit reveals that using only three FG transistors as shown in Fig. 16(b) allows us to remove mismatch induced variations in the f-i curve and the current threshold for repetitive spiking (bifurcation current). Firstly, to control the nullclines, the ability to tune the half-activation voltages of both the gating amplifiers is needed. Hence, M1 and M4 need to be floating-gates. This also makes the gain of the amplifiers relatively insensitive to mismatch as they are now set by capacitor ratios. As shown in Fig. 21(c), M_L sets the threshold current and hence needs to be a FG transistor. Variations in threshold of M_{Na} only shift the x-nullcline up or down in the region C, a phenomenon whose effect is nullified by ensuring that the n-nullcline does not intersect the x-nullcline in that part (by keeping k_amp_vss low enough).

The remaining two transistors, M_k and M3 and the capacitor C_k affect the slope of the f-i curve directly. To understand their effect, we need to analyze the components of a spike interval. The spike itself takes much less time compared to the time for the membrane to charge from its reset voltage, V_{reset} to the threshold for a spike V_{thresh}. For a simplified analysis, we can consider the time period to be determined by linear charging of a membrane capacitor by the difference between input and leak currents. This is of course a simplification as we ignore the ohmic nature of the transistors, but it does allow us to capture the trend easily. V_{thresh} is set by the location of zone B, i.e. the half-activation potential of the sodium channel. Hence, the time period of

oscillation for a certain input current is given by:

$$T \approx \frac{C_{mem}(V_{thresh} - V_{reset})}{I_{in} - I_L} \tag{15}$$

where V_{reset} is the voltage the membrane is reset to after a spike. V_{reset} increases for an increase in $V_{th,k}$ or a decrease in $V_{th,3}$. Figure 25 shows the simulated effect of a 30 mV increase in $V_{th,3}$ compared to the nominal case and the resulting variation in the f-i curve. From (15) it is obvious that increasing V_{thresh} appropriately can correct the slope of the f-i curve. Increasing V_{thresh} would, however, result in a change in the bifurcation current or the origin of the f-i curve. The leak current is then changed to bring the bifurcation current close to the desired value. The corresponding corrected curves are also plotted in the figure. Another possible method for eliminating mismatch in the slope of the f-i curve in a network of these neurons is modifying the synaptic efficacies of FG synapses connected to this neuron. In other words, synapses connected to different neurons will be of different baseline weights to produce the same result. The leak current needs to be modified in this case too.

Having shown that the three FG transistors in Fig. 16(b) can eliminate mismatch if their charge is modified appropriately, we need to estimate the sensitivity of the neuron's response to the programmed charge. Hence, the effect of finite resolution in programming the floating-gate charge is discussed next. We have earlier shown the ability to program FG currents to a 3-σ accuracy of 0.1% over 3-4 decades of current [8] or to 1% over 6-7 decades [15]. An accuracy of 1% in currents corresponds to an accuracy of 0.36 mV assuming subthreshold operation and assuming that threshold mismatch is dominant among all the sources of variation. To consider an extreme worst case scenario, we considered a 1 mV standard deviation in threshold voltages and performed a Monte Carlo analysis on the neuron's f-i curve. For each value of input current, 100 runs were performed to obtain mean and variance of the spike frequency. EKV models were used for the simulation. The results are plotted in Fig. 26. The resulting variation in the f-i curve is acceptable keeping in mind that this is

a conservative estimate.

3.4 Conclusion

Spurred by an increasing interest in real-time simulation of spiking neural networks, many researchers have developed integrated circuits modeling neurons over the last few years. These neurons have varied in detail from full Hodgkin-Huxley (H-H) [89, 95] to simple integrate and fire [51] models. The full H-H implementations have mostly suffered from excessive power dissipation and area penalties while integrate and fire models often do not possess many important dynamics (e.g. realistic phase response curves leading to synchronization). This has led researchers to develop hybrid models which strike a compromise and mimic certain properties that are most desirable to the user in their application. Spike frequency adaptation, refractory period, positive feedback and conductance based models are common examples of such properties [5, 38, 67]. Common implementation methodologies for these neurons are based on switched capacitors or current-mode design. While switched capacitor based methods offer good matching, they suffer from large area and limited programmability. Current-mode designs on the other hand are limited by mismatch and are difficult to bias. A comparison of several representative designs are presented in Table 3. The power consumption for a biological neuron is an approximate number obtained from [91] assuming that the 20 W of power dissipated in the brain is divided equally among its thousand billion neurons.The figure of merit (FOM) quoted in the table is a metric describing the complexity of the model, its power dissipation and normalized area. It is given by the following equation:

$$FOM = \frac{L_{min}^2.D}{A.P_{diss}} \tag{16}$$

where L_{min} is the minimum channel length in the VLSI process used, A is the area of the neuron, P_{diss} is the power dissipation for a spike rate of 100 Hz and D is a measure of the model detail. For different applications, some of these metrics (e.g. power)

may be more important than others; then that factor may be weighed more to arrive at a desired FOM. 'D' can be chosen in different ways depending on the application. In this case we wanted to use this to represent the variety of dynamical behaviors exhibited by the model. Hence we chose it to be a fraction which is determined by the number of features out of the twenty listed in [54] that is exhibited by that particular model. It should be noted that the detailed physics of an ion-channel or biologically realistic spike shape might be extremely important in some applications (e.g. dynamic clamps); this feature has not been taken as a metric here.

We have presented a bifurcation based silicon neuron that consumes the lowest power among all designs reported till date. Since it is based on a bifurcation, it possesses all the phenomenon/properties associated with the bifurcation. The proposed design mimics the dynamics of a type-I membrane and exhibits refractory period, positive feedback, and spike frequency adaptation. We also propose a circuit to extract the nullclines for the system which leads to an algorithm for correctly biasing the circuit. Combined with the ability of floating-gate transistors to set bias voltages locally, this method should reduce variation in silicon neurons while not incurring a significant area penalty.

To fully exploit the capabilities of this neuron in modeling biology, it has to be part of a network which should be specified by the user. The user should also have the freedom of choosing the parameters of the neuron. Realistic models of synapses and dendrites should also be available in large numbers in the system. For allowing large scale reconfigurability in a modular fashion, the structure of a FPAA described in [108] seems to be a logical step. We shall next describe how such a structure is adopted for our purpose and the features that have been added in the new version of the chip that I have developed.

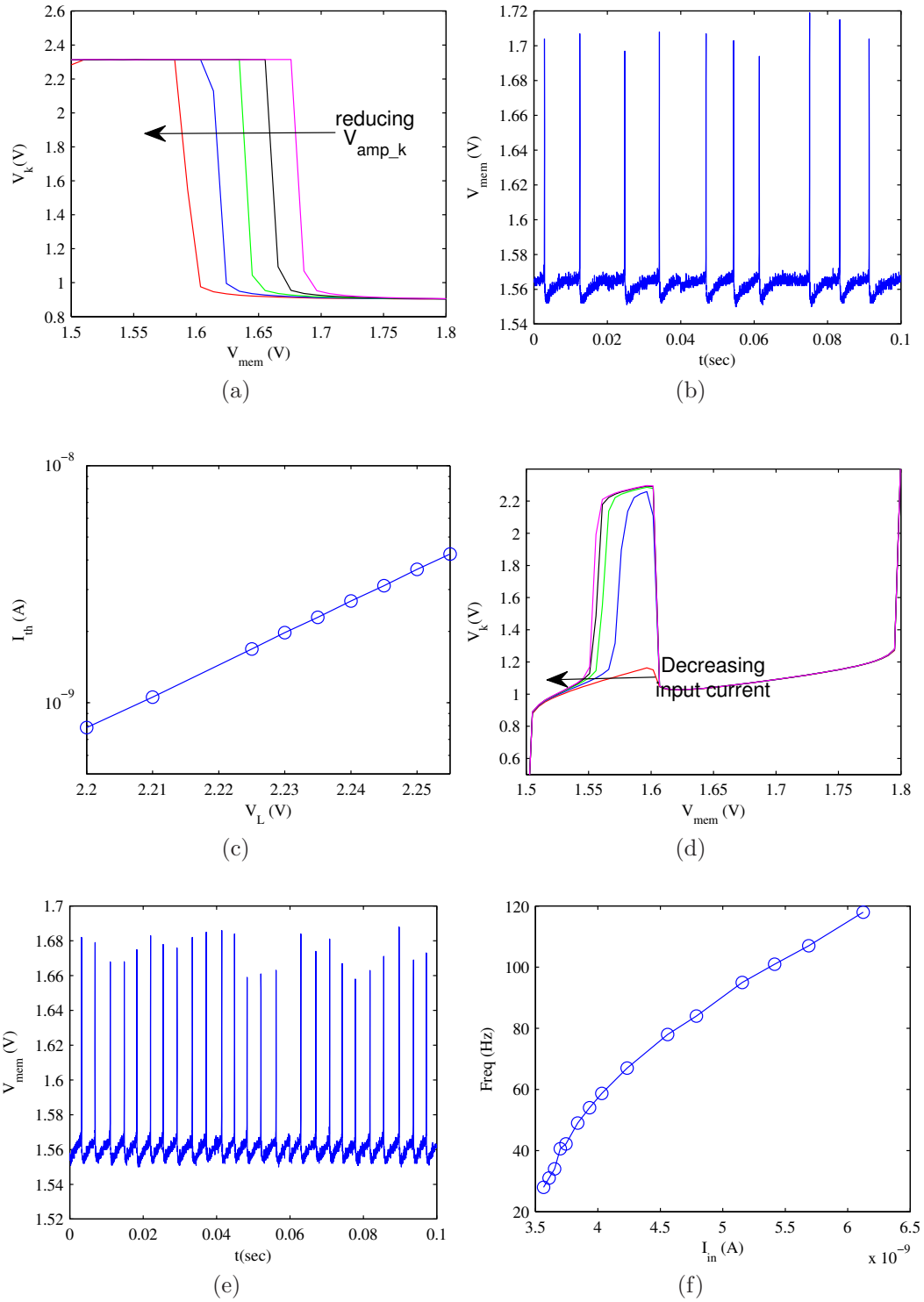


Figure 21: **Measured results:**(a) Variation of the n-nullcline with decreasing V_{amp_k} shows a decrease in o_2. (d) Increasing the input current stimulus resulting in variation of the x-nullcline. The existence of a saddle-node bifurcation can be inferred from the disappearance of the sharply rising part of the nullcline at the end of zone A in Fig. 17(a). (b,e) Measured spiking waveforms from the neuron for two different input current values. Absolute voltages are not shown. (c) The threshold current for repetitive spiking (bifurcation value) can be directly controlled using V_L. The resulting variation is exponential. (f) Measured f-I curve for the neuron.

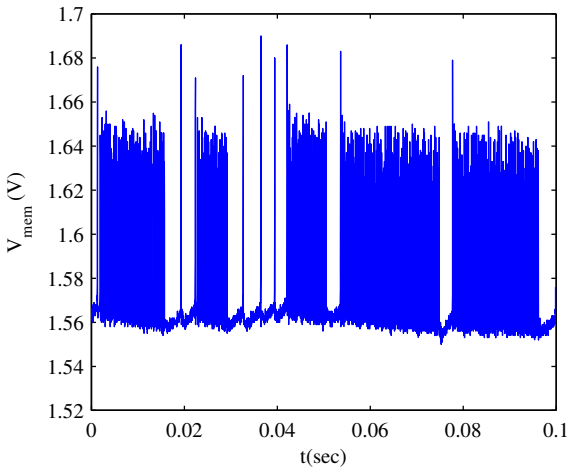


Figure 22: **Saddle Homoclinic bifurcation:** Keeping the input current lesser than the saddle-node bifurcation value and increasing $V_{\tau n}$ results in a saddle homoclinic bifurcation. The ambient noise switches the solution between the equilibrium and the limit cycle. Absolute voltages are not shown.

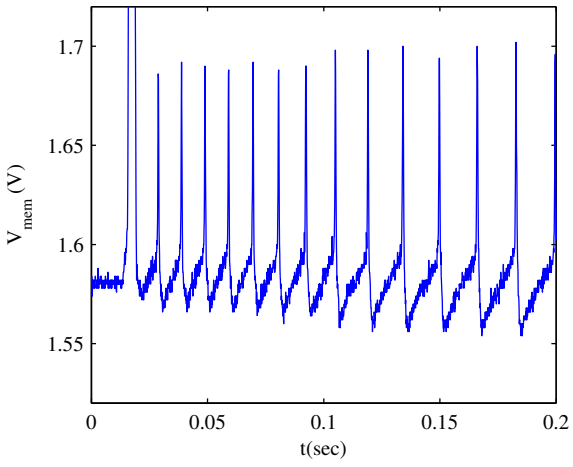


Figure 23: **Spike frequency adaptation:** Adding an extra potassium channel that has slower kinetics than the one responsible for regular spiking results in spike frequency adaptation. The measured waveform is the neuron's response to a current step.

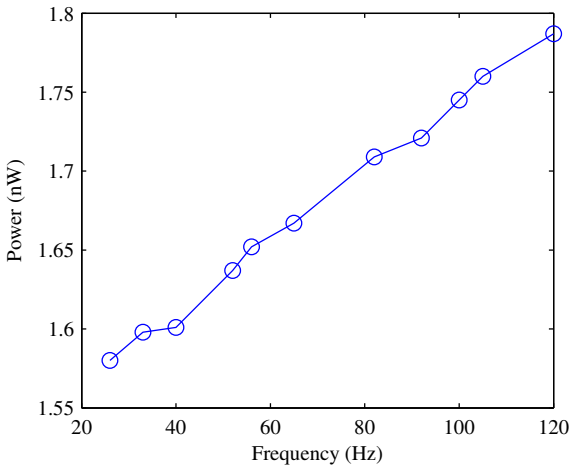


Figure 24: **Power dissipation:** Measured power dissipation of the neuron with spike frequency adaptation as a function of spike frequency. Its power consumption of 1.74 nW at a spike frequency of 100 Hz is the lowest reported till date.

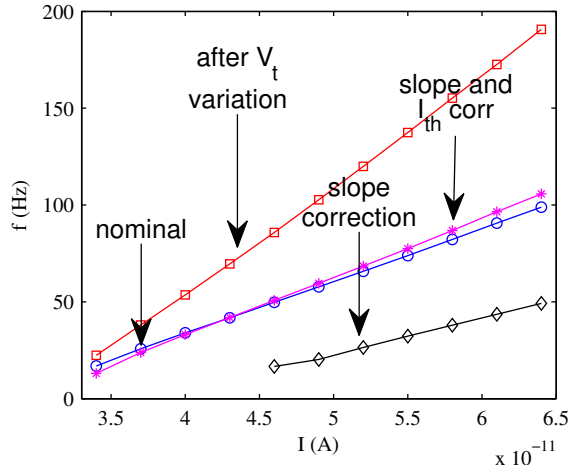


Figure 25: **Mismatch correction:** Threshold voltage variation of 30 mV in M3 causes a large change in the f-i curve. The slope can be corrected by changing the either the voltage threshold or the synaptic efficacy. The current threshold for repetitive spiking is modified by changing the leak current.

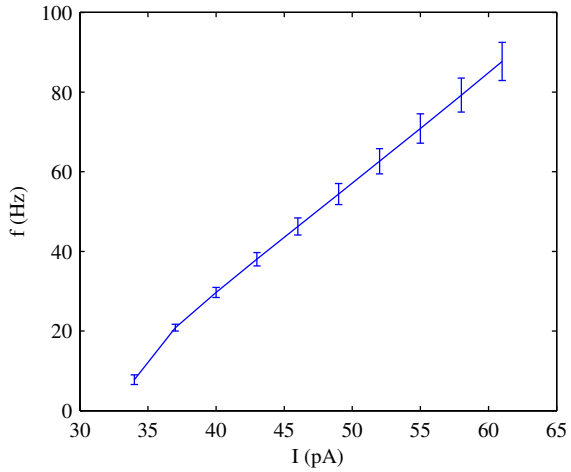


Figure 26: **Monte-carlo analysis:** A monte-carlo analysis is performed on the neuron circuit assuming a standard deviation of 1 mV for the transistor threshold voltages (assuming FG transistors are used to correct for the large threshold variations). The resulting f-i curve is plotted. EKV models were used for the simulation.

CHAPTER IV

FROM NEURONS TO NETWORKS: FIELD PROGRAMMABLE ANALOG ARRAY

In making the transition from single point neuron models to networks with complicated dynamics, the two important requirements are flexibility in defining the network topology and neuron parameters and incorporating high density of connections. A field-programmable analog array (FPAA) is suitable for this purpose. In fact, its architecture is suitable for a more general flexible analog processing paradigm since different applications can be targeted by modifying circuit blocks onle while keeping the general infrastructure intact. In this chapter, generic details of such a system are described while two specific examples are presented in the following chapters.

We present an FPAA with over fifty thousand floating-gate (FG) elements allowing it to operate as a reconfigurable analog processor. The FG devices serve as switches for reconfiguration and also endow the sub-circuits with tunable parameters. Using FG devices as switches eliminates the need for extra memory to store the configuration of the switches. Essentially, these devices perform 'computation in memory' leading to high computational area efficiency. This is one of the first large scale FPAA devices reported that is capable of versatile computations, making it a viable platform for prototyping and implementing varied processing tasks.

In the recent past, FPAAs have been gaining popularity because of their ability to allow rapid prototyping, flexibility in testing and reducing the design cycle time, reasons that led to the growth of the FPGA market. However, most of the FPAAs reported in literature or available commercially have a small number of computational analog blocks (CABs) with high performance components. The few designs

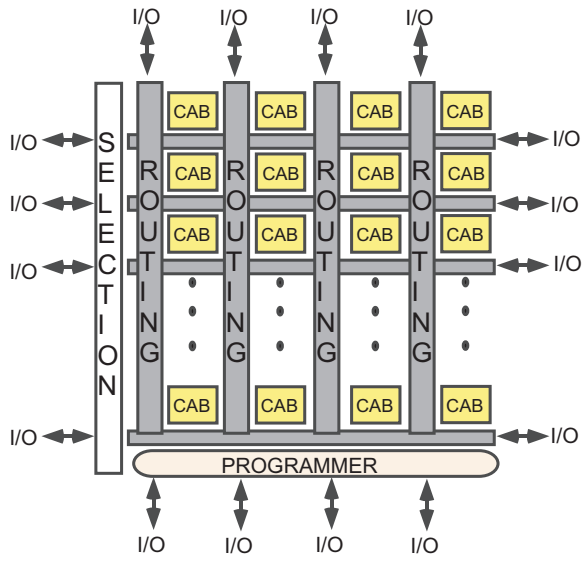


Figure 27: **Architecture:** The chip is organized into an array of 4X8 CABs with multi-level interconnect matrix in between. The circuitry for selection and programming of FG elements is at the periphery of the array.

that have a large number of CABS [16, 76] are specifically designed for constructing programmable analog filters. The chips we present has a a wide assortment of components in the CABs within one chip and have a different class of CAB components in different chips to better suit a separate application. This offers the user more flexibility in creating circuits. This also permits the usage of the chip in teaching analog circuits in a class, a task that has been performed multiple times.

An aspect of FPAAs that has attracted considerable attention is the parasitic effects and under-utilization of the switch matrix. A method for overcoming this problem is to use the switch matrix devices as valid circuit elements [16, 66]. In our chip too, since the voltage on the gate of the FG devices can be programmed in a continuum, we have used these devices as circuit elements in various cases. To reduce the parasitic capacitance due to the OFF switches in a crossbar type network, routing lines of varying sizes are present so that the length of the line chosen can depend on the number of components to be connected. Here, we present complete characterization of the architecture of the system in the following sections and also detail the software interface for configuring the chip.

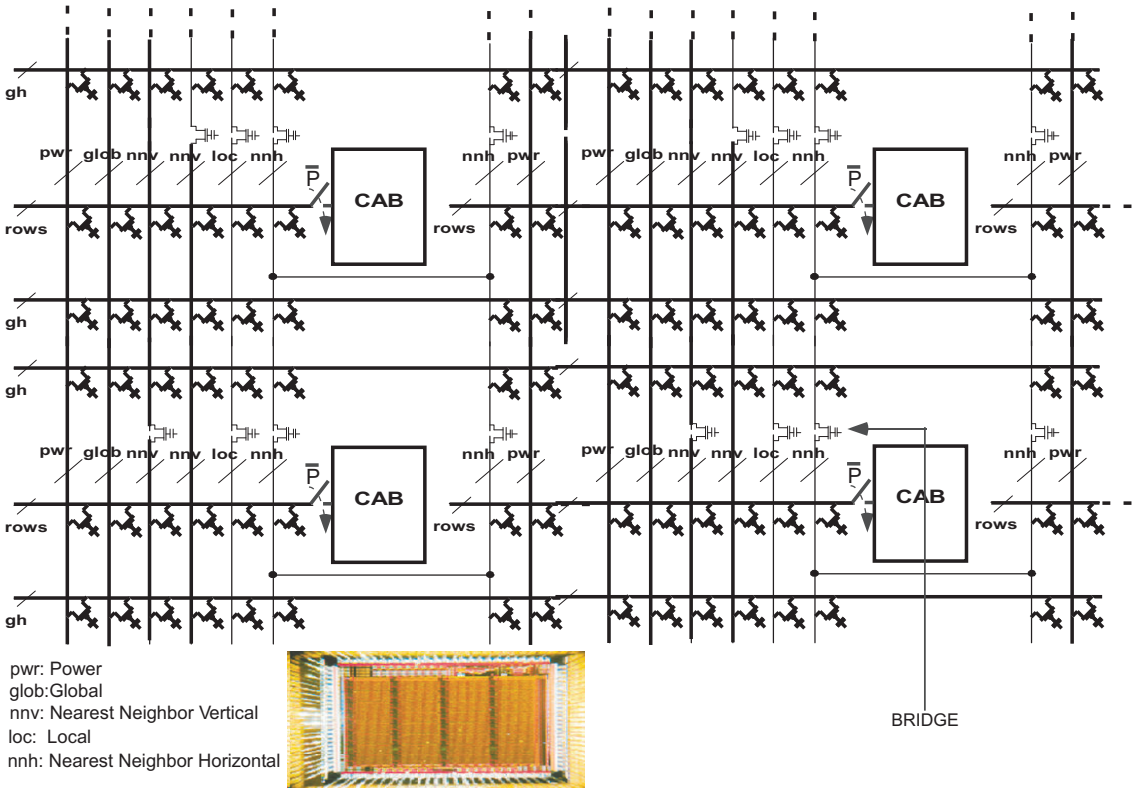
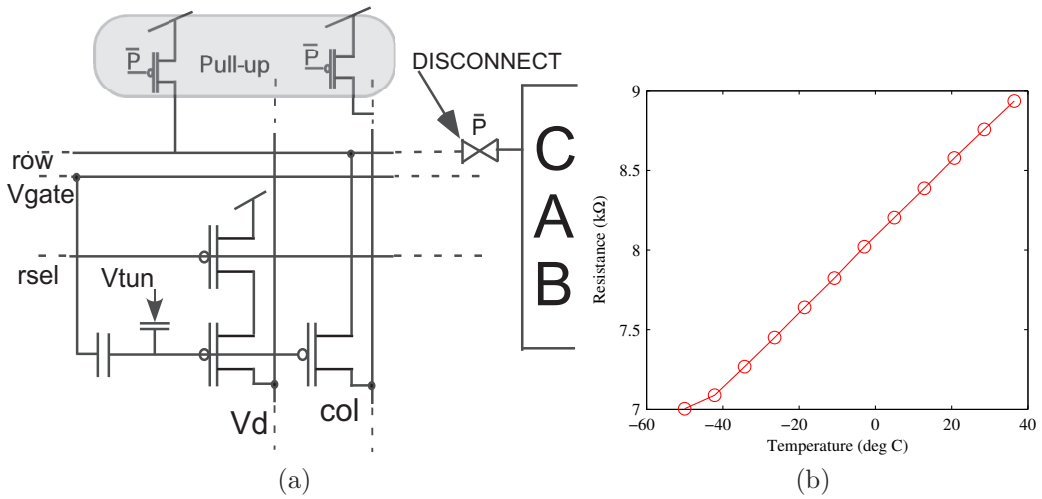


Figure 28: **Switch matrix:** The different categories of routing lines interconnecting the CABs is shown. Global, nearest-neighbor and local routing options are available on this chip allowing one to optimally route their circuit.

4.1 Architecture

The architecture of the chip is depicted in Fig. 27 and a die photo of one FPAA chip is shown in Fig. 28. It consists of a set of 32 CABs arranged in a 4X8 matrix. An interconnect matrix comprising FG elements allows for arbitrary connectivity between components. For the neural network chip, different neuron models have be placed in the CABs while the high-density components (e.g. synapses) have been implemented on the switch matrix. The selection and programming circuits for setting the desired charge on the FG devices are placed at the periphery of the array. The chip has 56 pins available for routing signals. More details about each part are presented in the following subsections.



(a) (b)

Figure 29: **Switch Performance:** (a) Architecture of the indirectly programmed switch and associated selection circuits is shown. (b) Switch resistance is plotted as a function of temperature. Its value at room temperature is around 12 $k\Omega$.

4.1.1 Routing

The routing architecture of the IC shown in Fig. 28(a) demonstrates the different types of interconnections-local vertical (*loc*), local horizontal (*rows*), nearest neighbor vertical and horizontal (*nnv* and *nnh*), global vertical (*glob*) and global horizontal (*gh*). The *glob/gh* lines span the entire length of the chip vertically/horizontally. There are 4 *gh*/10 *glob* lines per row/column of CABs. The local lines span the length/width of one CAB only. There are 41 *rows* and 4 *loc* lines. The *nnv* lines (6 for each neighbor) connect to vertically neighboring CABs while the *nnh* lines (4 for each neighbor) connect two horizontally neighboring ones. This granularity allows for high speed interconnects to be routed on low-capacitance lines like local or nearest neighbors while global connections are used only for I/O after the internal processing is complete. Bandwidth of a signal passing through four switches and two *nnv* lines has been found to be greater than 30 Mhz, the limit in that case being the signal generator. The other feature of the routing scheme is bridge transistors that allow *loc/nnv/nnh* lines to be bridged between CABs facilitating variable length connections without incurring the capacitance penalty of global lines.

The parasitic capacitance associated with the routing lines have also been estimated. A wide input linear range operational transconductance amplifier (OTA) was biased for a certain transconductance and step responses were measured with different routing lines as its load capacitor. A voltage buffer is used to isolate the pad capacitance from the Gm-C element. The resulting extracted capacitances are 1.6 pF for *glob*, 1.5 pF for *gh*, 552 fF for *nnh*, 458 fF for *nnv* lines and 220 fF for a *loc* line. Thus routing between CABs can be accomplished with relatively lower parasitic as compared to the earlier version. In addition, this characterization allows one to use the routing as extra capacitance that can be used in circuits.

4.1.2 Switch Isolation and Programming

The programmable switch matrix used in the earlier FPAA [108] used the application of high gate voltage or high drain voltage as the method of isolation while the selected device had a low voltage at both the gate and drain terminals. However this method has a number of disadvantages, the primary one being over-injection of devices beyond the isolation point [109]. This IC employs the superior source-side selection [109] coupled with indirect programming [42] to achieve impressive isolation while not sacrificing the quality of an ON switch. Figure 29(a) shows the architecture of one switch element that occupies 13×6 μm^2. The rsel signal provides source side selection by cutting off the current in the switches not being programmed. Vgate and Vd are the voltages provided by two on-chip DACs for programming the FG device. The pull-up transistors ensure that the direct FG device has its terminals at V_{dd} during 'program' mode so that it presents a similar capacitance as the indirect device would in 'run' or operational mode. This results in better matching between the two device currents. A similar architecture is used for the bias FG elements. For more precise matching between the direct and indirect transistors, an array of coefficients for V_T mismatch of each direct-indirect pair of transistors needs to be stored for predistortion

of the programmed currents.

The 'disconnect' switch serves to cut off the rows from the CAB based on a control signal. Each of these transmission gates have been found to provide 400 fF of capacitance when turned ON. This factor combined with maximum accurately programmed currents of 20 μA provides a severe limitation on the maximum achievable frequency on this FPAA and will be rectified in the next version.

Figure 29(b) shows the variation of the resistance of an ON switch with temperature. It increases with temperature due to decrease in mobility. It has a value of around 9 $k\Omega$ at room temperature. The OFF switches have a resistance that is more than 10 $G\Omega$. The ON switch resistance can be reduced further if the gate voltage (currently at mid rail) can be lowered to ground. This option is also being included in the next version of the chip.

4.1.3 On-chip Programming

Arguably, the most significant feature of this FPAA compared to earlier versions is its ability to program the FG currents over a wide dynamic range starting from very low currents. For biological applications, the time constants needed are very long ($\approx$ 100 msec - 1 sec). To attain these time constants without placing large capacitors on the chip, accurate programming of low currents is a necessity. The problem of measuring a wide dynamic range of currents is quite generic and hence we devote a couple of chapters in this thesis to discuss these issues. Hence, I just mention a few salient points about the FG programming infrastructure in this section and postpone a detailed analysis for later chapters.

The huge improvement in measuring accuracy, speed and dynamic range is obtained by using a logarithmic trans-impedance amplifier (TIA) described in [14]. SNR is improved by using source degeneration on the feedback PMOS. The TIA is kept

stable over a wide range of currents by adaptively biasing it based on the input current. In this implementation two such logarithmic current measurements are used, one for higher currents and the other for lower ones. This structure has allowed us to measure currents as low as 100 fA while earlier off-chip measurements saturated at around 100 pA due to ESD leakage and noise.

Another important improvement in programming is introduction of row-parallel programming for switches. The rows of the floating-gate array are selected by a decoder but the columns are selected using a shift register which enables selecting multiple columns per row. This leads to switch programming time given by $N_{rows}\times$ 1 msec.

4.1.4 I/O pad and Scanner Shift Register

Special bidirectional I/O pads have been incorporated into this IC which have buffer amplifiers capable of driving high capacitive loads when enabled. Their bandwidth is determined by a programmable floating gate device. For a total bias current of 2.1 mA (including biasing circuits), the corresponding amplifier (designed following [32]) achieves an unity gain bandwidth of 48 Mhz, a DC gain of 103 dB and a phase margin of 70° while driving a capacitive load of 15 pF in SPICE simulations. Also an analog 16 bit shift register is available to scan through and observe different lines allowing the user an option to debug their circuit almost in a SPICE-like fashion.

4.2 Software Interface

Configuring the FPAA to implement a particular circuit requires turning ON a certain set of switches and accurately programming some FG elements for biases. A software tool chain has been developed to enable this task. Figure 30 shows a graphical description of the software flow. A library containing different circuits (neuron, dendrite, transconductor, multiplier etc) is used to create a larger system in Simulink,

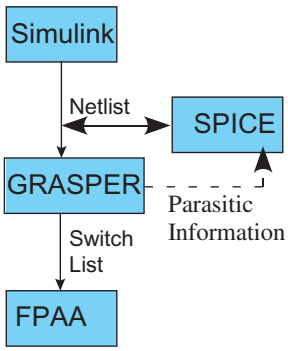


Figure 30: **Simulink:** Flow of the software infrastructure is shown. A simulink level description is first translated to a Spice netlist and then converted to a list of switches on the FPAA. The tool for mapping a netlist to the FPAA also provides information on routing parasitics that may be used to manually tune the parameters.

a software product by The Mathworks. The circuits in the library correspond to pre-constructed SPICE sub-circuits whose parameters can be set through the Simulink interface. A first code (sim2SPICE [82]) converts the Simulink description to a Spice netlist, while a second one (GRASPER [10]) compiles the netlist to switch addresses on the chip. A Spice file can also be a direct input to GRASPER. The conversion of netlist to FPAA switches involves two parts - placement and routing (similar to a digital VLSI flow). The algorithm for placement tries to maximize the possibility of local wire usage while the routing algorithm is based on a maze routing approach [64] based on Djikstra's shortest path algorithm. GRASPER also provides a parasitic annotated post-routing netlist that can be used to re-tune the circuit parameters for desired performance. This feedback is currently manual but its automation is being implemented.

Figure 31(a) shows an example of a Simulink level circuit description while Fig. 31(b) shows the schematic of the circuit. The vector matrix multiplier (VMM) [24] enables multiplying a vector of input signals with a matrix of fixed coefficients/weights and will be described in detail later in the thesis. The schematic shows a 2-input 1-output VMM with the output given by:

$$I_{out} = w_1 I_{in1} + w_2 I_{in2} \tag{17}$$

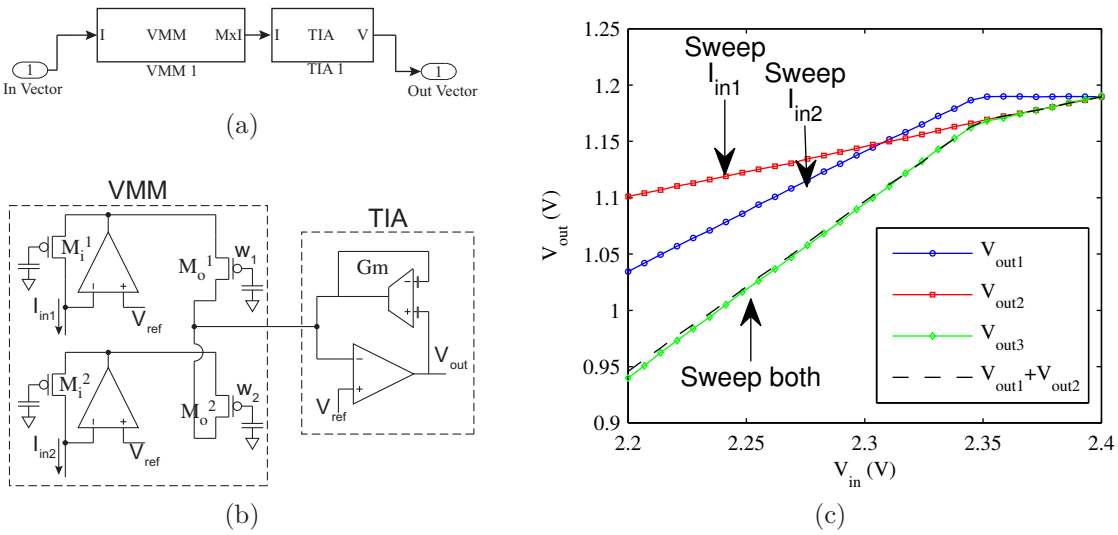
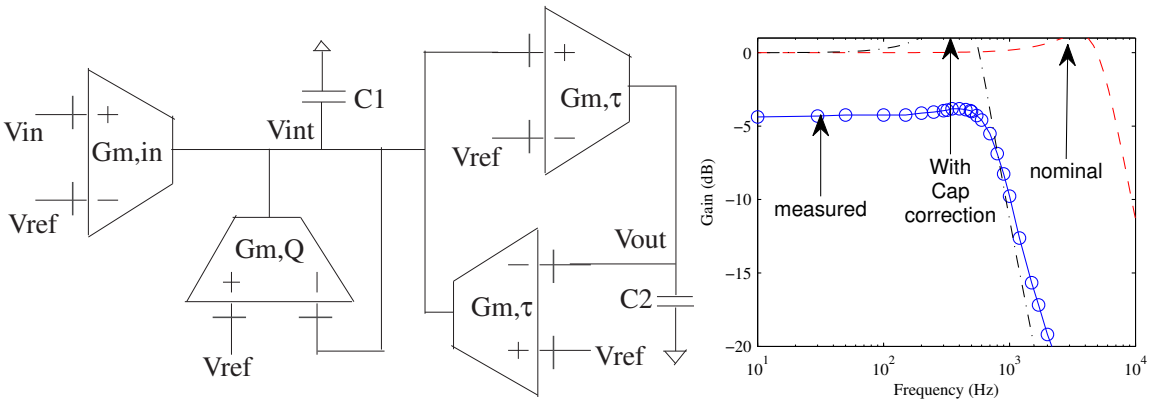


Figure 31: **Simulink example:** (a) Block diagram description of a vector matrix multiplier (VMM) followed by a TIA. (b) Circuit schematic of the single ended VMM followed by a TIA where a wide linear range transconductor is used as a resistor. (c) Measured data from the FPAA showing the input multiplied by different weights.

where w_1 and w_2 are set by difference in charge on the gates of the M_i and M_o transistors. This is the simple case where the weight matrix is just a vector and the output is the projection of the input vector on the weight vector. The output currents are converted to a voltage using a transimpedance amplifier (TIA) where the resistor is implemented using an floating-gate OTA (FGOTA) resulting in $R = 1/G_m$. The input currents are also created using FGOTA circuits. Figure 31(c) shows the output of the TIA plotted against the input voltages applied to the FGOTA elements to create I_{in1} and I_{in2}. The two different slopes representing w_1 and w_2 are obtained by sweeping only one input. Sweeping both the inputs results in a slope that is nominally equal to $w_1 + w_2$. The offset in the curves results from FGOTA offsets which can be corrected (shown later).

Figure 32(a) shows the circuit of a second-order low-pass filter based on a resonator topology. FGOTA elements are used here to increase linear range. This circuit is used to demonstrate the parasitic capacitance prediction feature. The transfer function of



(a) (b)

Figure 32: **Gm-C filter:** (a) Circuit diagram for a second order low-pass filter based on a resonator topology. (b) Measured corner frequency from the FPAA matches well with theory simulation obtained after including the effect of parasitic capacitances and G_m mismatch.

this circuit is given by:

$$H(s) = \frac{V_{out}(s)}{V_{in}(s)} = \frac{G_{m,in}/G_{m,\tau}}{1 + \frac{sC_2G_{m,Q}}{G_{m,\tau}^2} + \frac{s^2C_1C_2}{G_{m,\tau}^2}}$$
$$f_0 = \frac{G_{m,\tau}}{2\pi\sqrt{C_1C_2}}, Q = \sqrt{C_1/C_2}\frac{G_{m,\tau}}{G_{m,Q}} \quad (18)$$

In this implementation, both $G_{m,\tau} = G_{m,Q} = 15.5$ nS while $C_1 = C_2 = 0.5$ pF resulting in Q = 1 and $f_0 = 4.95$ kHz. Figure 32(b) plots the theoretical frequency response of the circuit for these nominal values. However, the measured frequency response has a smaller $f_0 = 850$ Hz because of parasitic capacitances. The measured Q is 1.05 which is close to the expected value. Using the back-annotated parasitic values obtained from GRASPER, the theoretical value becomes $f_0 = 480$ Hz, which is close to the measured value. Thus, using these parasitic capacitance values the G_m values can now be modified to get the desired frequency response. It should be noted that the Q-mismatch is because of a mismatch in the measured G_m values due to the mismatch between the indirect programming transistor and the direct transistor [42] in the circuit. This can be corrected iteratively or by characterizing and storing this mismatch for every FG device on the chip.

In the following sections, two different chips are described that I have designed for spiking neural networks and analog signal processing. In all the experiments, the

Figure 33: **Printed circuit board:** Test setup for the chip includes a microprocessor, voltage regulators and data converters on the same board. USB is used for communication with the PC and also for the board's power supply.

power supply of the chip is maintained at 2.4 V. A printed circuit board, shown in Fig. 33 has been designed for testing and demonstration. It has a microprocessor for programming and control, regulators for power supply and data converters in addition to the FPAA itself. USB connection serves as the communication link with the host PC and also provides power to the board. This makes the entire setup portable. In the next chapter we describe the chip designed for spiking neural network simulations.

CHAPTER V

RECONFIGURABLE ANALOG NEURAL NETWORK

The massive parallelism offered by VLSI architectures naturally suits the neural computational paradigm of arrays of simple elements computing in tandem. We present a neuromorphic chip with 84 bandpass positive feedback (e.g. transient sodium) and 56 low-pass negative feedback (e.g. potassium) ion-channels whose parameters are stored locally in floating-gate (FG) transistors. Hence, fewer but detailed multi-channel models of single cells or a larger number (maximum of 84) of simpler spiking cells can be implemented. Other components in the CABs also allow building integrate and fire neurons, winner-take-all circuits and dendritic cables making this chip a perfect platform for computational neuroscience experiments. Moreover, since this chip produces real-time analog outputs it can be used in a variety of applications ranging from neural simulations to dynamic clamps and neural interfaces as shown in Fig. 34.

Several systems have been reported earlier where a number of neurons were integrated on a chip with a dense synaptic interconnection matrix. Though these chips definitely accomplished the tasks they were intended for, large scale hardware systems modeling detailed neuron dynamics (e.g. Hodgkin-Huxley, Morris-Lecar etc) seem to be lacking. One attempt at solving this problem is presented in [89]. However, the implemented chip had only 10 ionic channels and 16 synapses, with a large part of the chip area devoted to analog memory for storing parameter values. Another approach reported in [115] had 4 neurons and 12 synapses with sixty percent of the chip area being occupied by digital-analog converters for creating the various analog parameters.

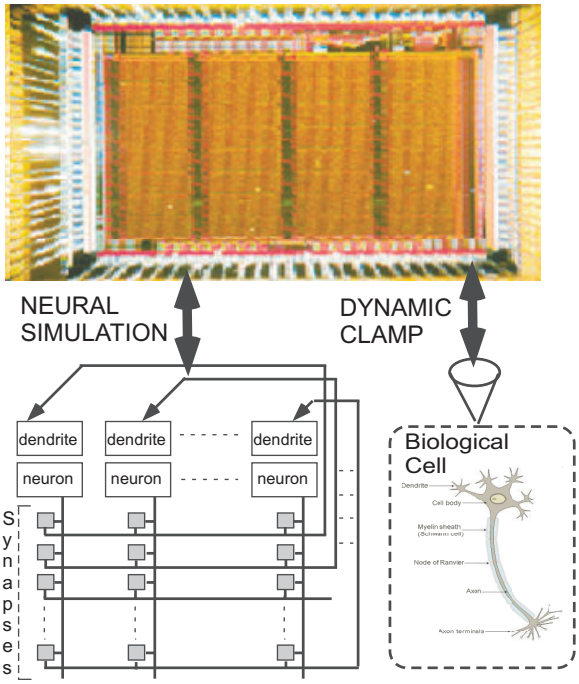


Figure 34: **Versatility:** Our neuromorphic chip can be used not only for simulating neuron populations but also for interfacing with live neurons in a dynamic clamp setup or for neural prostheses.

In the following sections, we describe the components in the CAB. Then we present measured data showing the operation of channels, dendrites and synapses. Finally, we show some larger systems mimicking central pattern generators or cortical neurons and conclude in the final section with some remarks about the computational efficiency, accuracy and scaling of this approach.

5.1 System Overview

5.1.1 Chip Architecture

Figure 35(a) shows the block level view of the chip shown in the earlier chapter. The first row of CABS has bias generators which can produce bias voltages that can be routed along columns for all the computational CABs. It should be noted that the regular architecture allows for tiling multiple chips on a single board to make larger modules. Figure 35(b) is a layout photo of the fabricated chip while Fig. 34 shows a die photo.

Figure 35(c) shows the components in the CAB. The components 1 and 2 in the

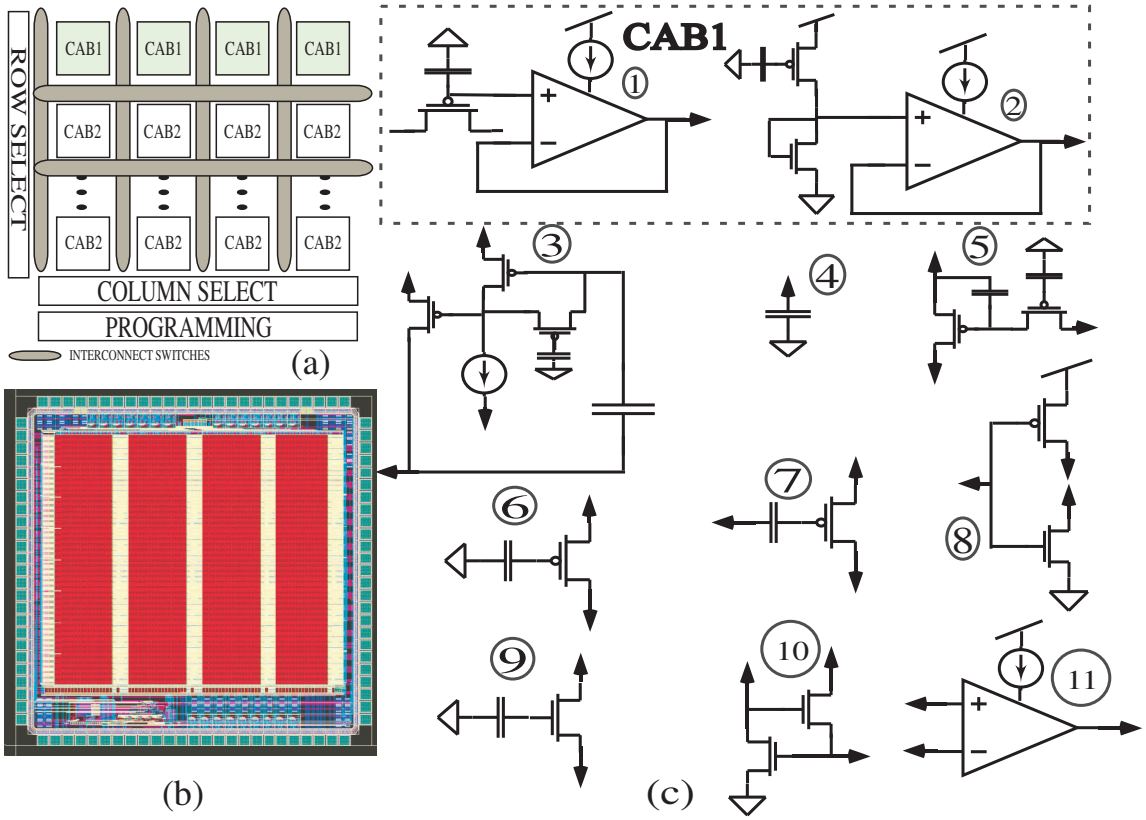


Figure 35: **Chip Architecture:** (a) The chip is organized into an array of 4x8 blocks that can be interconnected using floating-gate (FG) switches. The beauty of this architecture is the FG switches can be used not only for routing but also to set synaptic weights. (b) Layout of the chip fabricated in 0.35 μ CMOS. (c) CAB components that are used for computation along with the switch matrix elements. The tunneling junctions and programming selection circuitry for the floating gates are not shown for simplicity. The arrows on the components denote nodes that can be connected to other nodes through routing.

dashed square are in CAB1. In both the cases, the floating gates are programmed to a desired level and the output voltage is buffered using a folded-cascode operational transconductance amplifier(OTA). The bias current of the OTA can also be programmed allowing the amplifiers to be biased according to the application, thus saving power. As mentioned earlier, the CABs in the first row are of this type.

In CAB2, components 3 and 5 are the positive feedback and negative feedback channels respectively. In the context of Hodgkin-Huxley neurons, they are the sodium and potassium channels respectively. However from the viewpoint of dynamics, these blocks could represent any positive feedback (or amplifying [55])inward current and negative feedback (or resonant) outward current. Circuit 11 is a programmable bias

OTA which is included because of its versatility and omnipresence in analog processing. Circuit 4 is a 100 fF capacitance that is used to emulate membrane capacitance. Different magnitudes of capacitance are also available from the metal routing lines and OFF switches. One input of a current mode winner-take-all block is formed by 10. A synapse following the implementation in [41] can be formed out of components 6, 7, 8 and 9 and will be detailed later. The reason for choosing such a granularity is primarily component reuse. For example, component 8 can also be used as a variable current sink/source or a diode connected FET while component 6 can be used as a leak channel. We next describe how to combine different CAB components to make neurally inspired circuits.

5.2 Spiking Neuron Models

5.2.1 Hopf Neuron

Figure 36(a) shows the circuit for a Hodgkin-Huxley type neuron consisting of a sodium and a potassium channel. For certain biasing regimes, the neuron has a stable limit cycle that is born from a Hopf bifurcation, the details of which are available in [13]. In this case, we have biased the potassium channel such that its dynamics is much faster than the sodium (M3 acts as an ON switch). Hence, the potassium channel acts like a leak channel. The whole system now becomes a two-dimensional set of differential equations since the dynamics of V_{mem} follow that of the sodium channel. The parameters of the sodium channel are set based on voltage clamp experiments on it (not shown here).

It is important to understand that these neurons have different computational properties when compared with integrate and fire neurons. For example, the frequency of spikes does not reduce to zero as the bifurcation value is reached, a classical property of type II neurons [55]. Also, synchronization properties and phase-response curves of these neurons are significantly different from integrate and fire neurons. Hence,

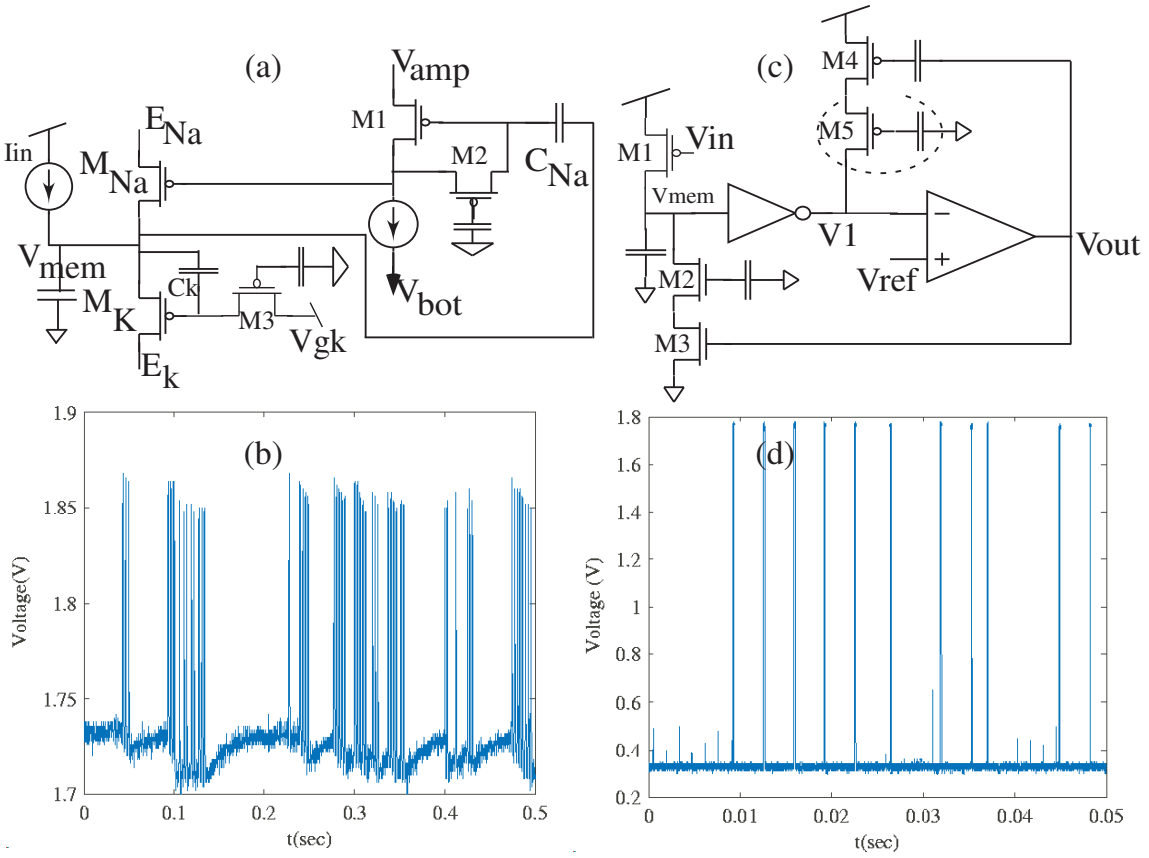


Figure 36: **Spiking Neuron:** (a) Neuron model where spiking is initiated by a Hopf bifurcation. (b) Measured noise induced spikes when the neuron in (a) is biased at the threshold of firing. (c) Integrate and fire neuron with the hysteresis obtained using M4 and M5. Here, the circled transistor, M5 is a switch element. (d) Measured noise induced spikes when the neuron in (c) is biased at the threshold of firing.

it is an indispensable component of a library of neuronal components. Figure 36(b) shows measured noise induced spikes from a Hopf neuron biased at the threshold of firing. Note that the magnitude of the action potentials are similar to biology, thus opening the possibility of using the chip for interfacing with live neurons.

5.2.2 Integrate and Fire Neuron

Figure 36(c) shows the circuit used for an integrate and fire neuron. The circuit has a hysteresis loop based relaxation oscillation when the input current is large enough. The inverter exhibits hysteresis because of the feedback from M4 and M5. M4 and M5 act as a current source, I_{hyst}, when V_{out} is low, while it is turned OFF when V_{out} is high. M5 is a routing element that sets the value of I_{hyst} while M4 acts as a switch.

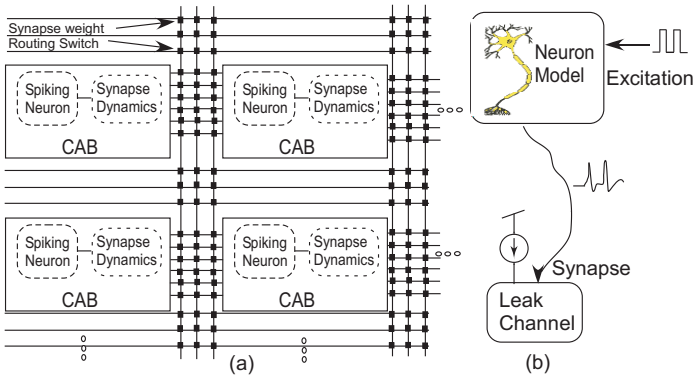


Figure 37: **Synapse architecture:** (a) The architecture of the chip places the synapse dynamics block for both excitatory or inhibitory synapses in the CAB along with the model of the soma. The synapse weight is set by the interconnect network. (b) The test setup for the experiment has an excitable neuron with a synaptic connection to a leak channel biased by a current source. A pulse is applied to the neuron and the corresponding PSP is measured. Compiling the circuit from the SIMULINK GUI is as easy as drawing this cartoon.

The trip-point of the inverter depends on the condition of V_{out} leading to hysteresis. The time period of the relaxation oscillations is given by:

$$T = \frac{V_{hyst}}{I_{in}} + \frac{V_{hyst}}{I_{in} - I_{reset}}, \tag{19}$$

where V_{hyst} is the magnitude of the hysteresis loop in terms of the membrane voltage and I_{reset} is the reset current controlled by M2 and M3. It can be seen that the frequency of oscillations in this case does reduce to zero as I_{in} reduces to zero, akin to a type I neuron. This system can also be modeled by a differential equation with two state variables. Figure 36(d) shows output of this circuit due to noise when it is biased at the threshold of firing.

5.3 Synapse

In this section, we describe three possible methods of implementing synaptic dynamics in the chip. The overall architecture is depicted in Fig. 37(a). Every CAB has a spiking neuron and a circuit to generate the dynamics of a post-synaptic potential (PSP). This node can now be routed to other CABs having other neurons. The FG switch that forms this connection is however not programmed to be fully ON. Rather

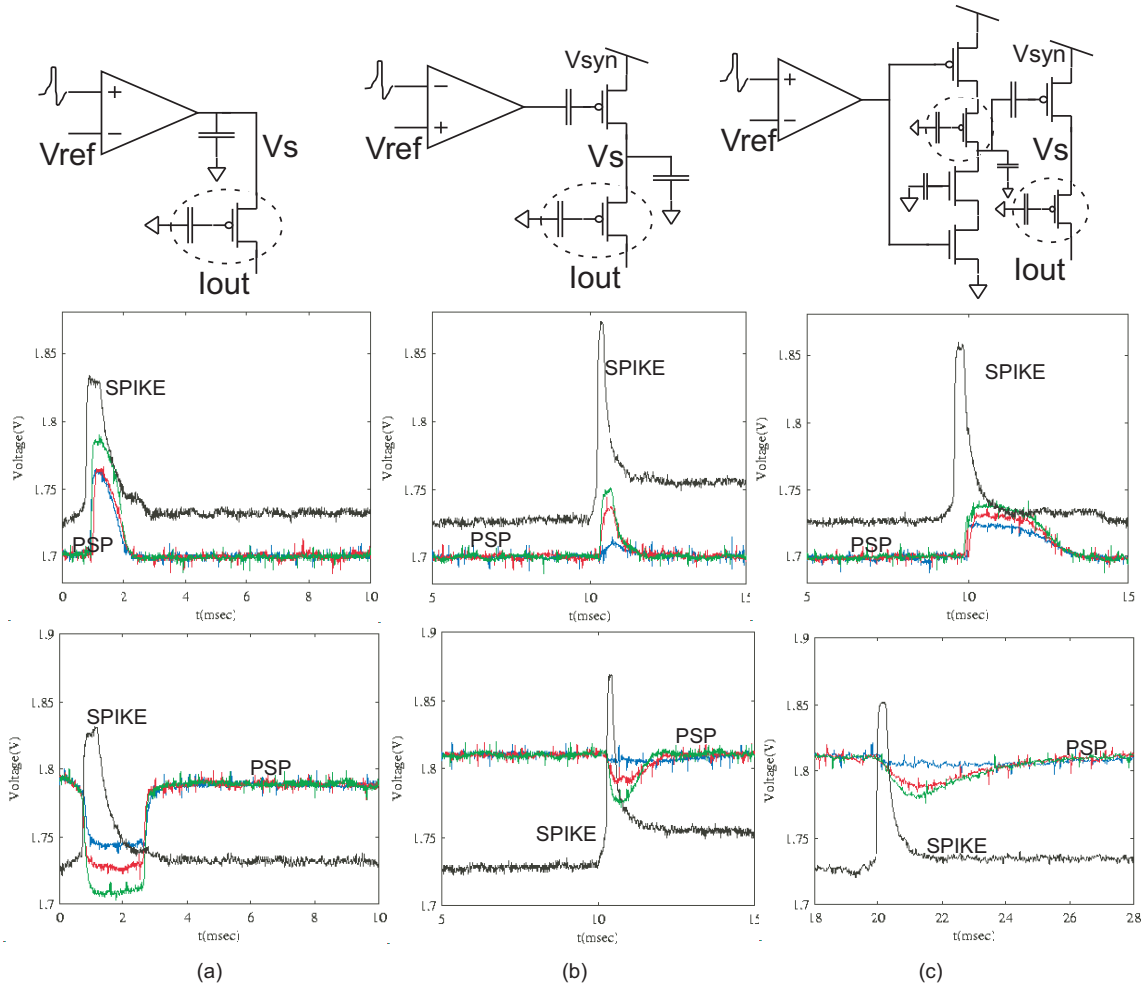


Figure 38: **Synapse:** Three possible chemical synapse circuits. The circled transistor represents a switch element. PSP for three different weight values are shown. (a) The simplest excitatory synapse where reversing the positive and negative terminals of the amplifier changes it to an inhibitory synapse. (b) The amplifier acts as a threshold and switches a current source ON/OFF. The value of V_{syn} relative to the membrane potential makes it inhibitory or excitatory. (c) Similar to (b) with better control on the shape of the PSP waveform because of the current starved inverter governing the charging and discharging rates independently.

the amount of charge programmed onto its gate sets the weight of this particular connection that is accurate to 9 bits. Hence, all the switch matrix transistors act as synaptic weights leading to all to all connectivity in the chip.

Figure 37(b) shows the setup for measuring the dynamics of the chemical synapse circuit. A neuron is biased such that it elicits an action potential when a depolarising current input is applied to it. This neuron has a synaptic connection to a passive

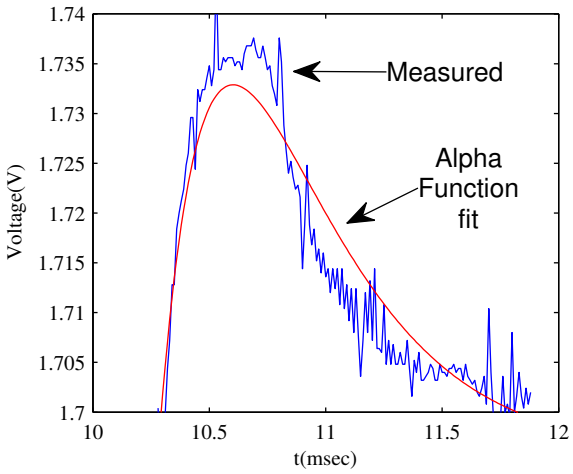


Figure 39: **Rall's alpha function:** Rall's alpha function is fit to one of the EPSP plots from the earlier experiment with a resulting error less than 10 %.

membrane with a leak conductance where the PSP is measured. Out of the many possible synaptic circuits possible, we show three versions here. All these circuits have the dynamics of a one-dimensional differential equation. Unlike these chemical synapses, electrical synapses are almost instantaneous and can be modeled by a floating-gate PMOS. We have also measured such circuits and their effects in synchronization of spiking neurons but do not discuss them here.

Figure 38(a) depicts the simplest type of excitatory synaptic circuit. The amplifier creates a threshold at V_{ref} and charges or discharges the node V_S when the input voltage crosses the threshold. Depending on the charge on the floating-gate switch element (circled transistor), a certain amount of current is then incident on the post-synaptic neuron. The synapse becomes inhibitory if the input is applied to the negative terminal of the amplifier. We show measured data for both cases for three different synaptic weights.

Figure 38(b) shows the second circuit for a chemical synapse and measured results for the same. Here, the amplifier creates a digital pulse from the action potential. This switches the floating-gate PMOS current source ON which charges the node V_S while the second FG routing element sets the weight of the connection. The synapse is excitatory when V_{syn} is larger than the resting membrane potential and inhibitory

otherwise.

Figure 38(c) shows the circuit which replicates the synaptic dynamics most accurately [41]. After the amplifier thresholds the incoming action potential, the current starved inverter creates an asymmetric triangle waveform (controlled by the FG PMOS and NMOS) at its output. The discharge rate is set faster than the charging rate leading to post synaptic potentials that decay very slowly. Again, we show EPSP and IPSP waveforms for three weights of the synapse. The waveforms shown are close to the ones recorded for actual neurons [59]. A common method for modeling PSP is using Rall's alpha function as follows:

$$V_{PSP}(t) = V_{max}\alpha t e^{(1-\alpha t)} \tag{20}$$

Figure 39 shows a curve fit of such an alpha function to a measured EPSP waveform with an error that is less than 10%.

It should be noted that when using these synapses with integrate and fire neurons, the amplifier used for thresholding is not needed as it is part of the neuron circuit.

5.4 Dendrite

The circuit model of a dendrite that we use is based on the diffuser circuit described in [45]. This is one of the circuits built entirely on the routing fabric and exploits fully the analog nature of the switches. Figure 40(a) shows the circuit while Fig. 40(b) shows a modified version used to measure the different branch currents. The horizontal transistors connecting the nodes V_x allow diffusion of currents while the vertical transistors leak current to a fixed potential from every node. The dynamics of an n-tap diffuser circuit is represented by a set of n-dimensional differential equations which approximate a partial differential equation. The steady state solution of the equation is exponentially decaying node currents as the distance of the node from the input node increases.

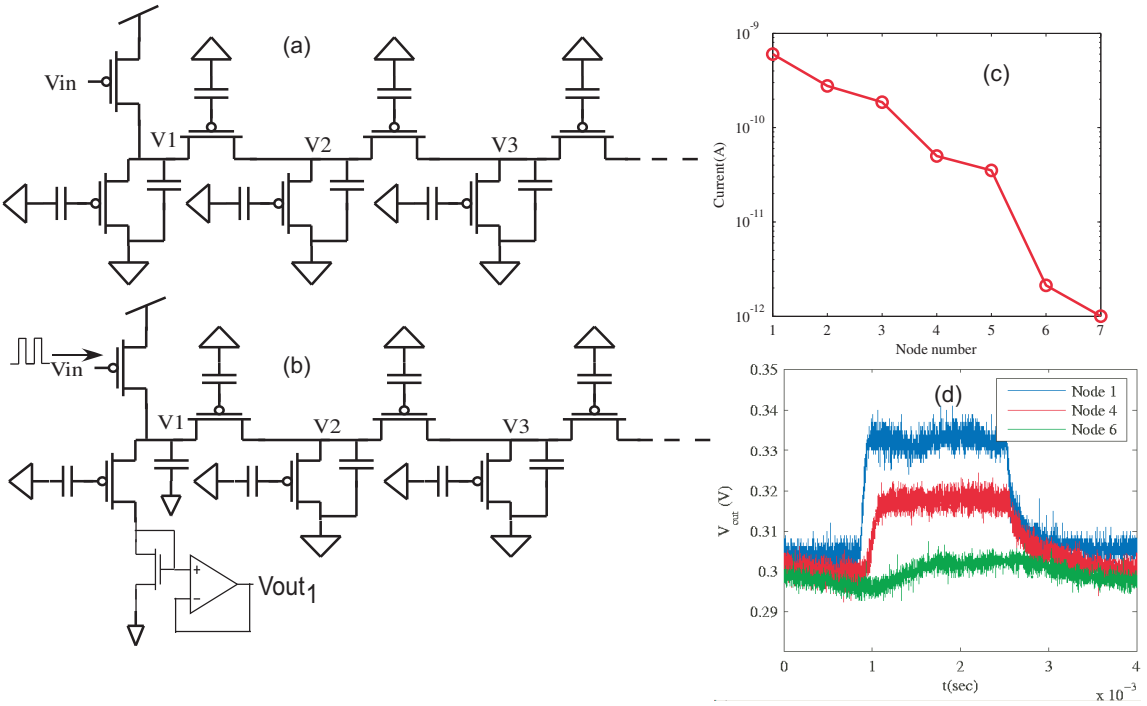


Figure 40: **Dendrite model:** (a) Model of a passive dendrite based on a diffuser circuit. (b) Experimental setup for measuring transient response of a dendrite cable. The current through the desired node is converted to a voltage using the diode connected NMOS. (c) Steady state currents in a 7-tap diffuser. (d) Step responses at nodes 1,4 and 6 of a 7-tap diffuser showing progressively more delay.

Figure 40(c) plots the steady state current through the compartments of a 7-tap diffuser. Figure 40(d) shows step responses of a seven tap diffuser. Voltages at the first, fourth and sixth nodes are plotted here. The delayed response of the distant nodes is typical of dendritic structures. The effect of the changing diameter in dendrites can also be modeled in these circuits by progressively changing the programmed charge on the horizontal devices along the diffuser chain.

We can put together all the previous circuit elements by creating a spiking neuron that has a synapse connecting it to the dendritic tree of another neuron. Figure 41(a) shows a picture depicting such an experimental setup, the results of which are presented in Fig. 41(b). We can see that initially the post-synaptic neuron does not respond to the input spikes. However, increasing the dendritic diffusion results in visible post-synaptic potentials. Increasing the diffusion even more allows the post-synaptic neuron to fire in synchrony with the pre-synaptic one.

5.5 Larger Systems

Spiking neurons coupled with synapses have been the object of considerable study over several years. While there are theories showing the existence of associative oscillatory memory [52] in networks of coupled spiking neurons, a lot of work has been devoted to looking at the simplest case of two coupled neurons and its role in generating rhythms for locomotion control [113]. The most popular circuit in this regard is the half-center oscillator where the neurons are coupled with inhibitory synapses. Here, we look at both cases, i.e. when the connections are inhibitory or excitatory as shown in Fig. 42(a). Intuitively, when the connections are excitatory both the neurons will try to fire at the same time, leading to in-phase spikes. On the other hand, when the connection is inhibitory, the spiking of one neuron suppresses that of the other giving rise to spikes out of phase. This phenomenon and its relation to synaptic strength can be studied better by transforming the differential equation into a phase variable. We can transform the equations using the moving orthonormal co-ordinate frame theory [27] and keep the first order approximation of a perturbation analysis to get:

$$\dot{\phi}_i = \alpha_i + \varepsilon \sum_{j \neq i} H_{ij}(\phi_i - \phi_j), \phi \epsilon S^1 \tag{21}$$

where ε is the synaptic strength and α_i are frequency deviations from a nominal oscillator. For two oscillators, it can be seen that for a given frequency deviation there is a fixed point only if ε is larger than a certain minimum value. In practice, no two spiking neurons have same frequency of oscillation even at same biasing because of mismatch. So, in experiments, we slowly increased the synaptic strength till the oscillators synchronized. Figure 42(b) and (c) show the measured spiking waveforms obtained from integrate and fire neurons and Hopf neurons respectively. We can see that in one case the neurons are spiking in-phase while they are anti-phase in the other. All these measurements were done with synapses of the first kind discussed

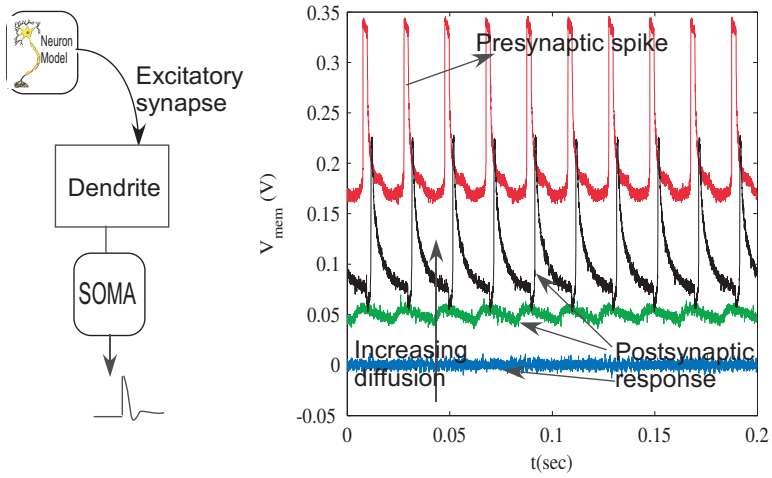


Figure 41: **Full Neuron:** (a) A spiking neuron is connected to another neuron through an excitatory synapse. The post synaptic neuron in this case has a dendritic tree. (b) The diffusion length of the dendrite is slowly increased in the experiment. Though the post synaptic neuron did not respond initially, increasing the diffusion resulted in visible EPSP waveforms and eventual spiking. Absolute voltages are not shown in this figure.

earlier. They can be done with different synaptic dynamics to analyze the effect of synaptic delay on synchronization properties of type I and type II neurons. Detailed dynamics of escape and release phenomenon [97] can also be observed.

Figure 43(a) shows the schematic for a central pattern generator for controlling bipedal locomotion [113] or locomotion in worm like robots [4]. It consists of a chain of spiking neurons with inhibitory nearest neighbor connections. We implemented this system on our chip with Hopf neurons connected with the simple synapses described earlier. The resulting waveforms are displayed in Fig. 43(b). The current consumption of the neuron in this case is around 180 nA and the synapse dynamics block consumes 30nA of current leading to a total power dissipation of around 0.74 μW (excluding power for biasing circuits and buffers to drive off-chip capacitances). The low power consumption of the computational circuits and biological voltage scales makes this chip amenable for implants.

The second system we present is a spiking neuron with four dendritic branches that acts as a spike sequence detector. Figure 44(a) shows the schematic for this experiment. In this experiment, the dendrites were chosen to be of equal length and

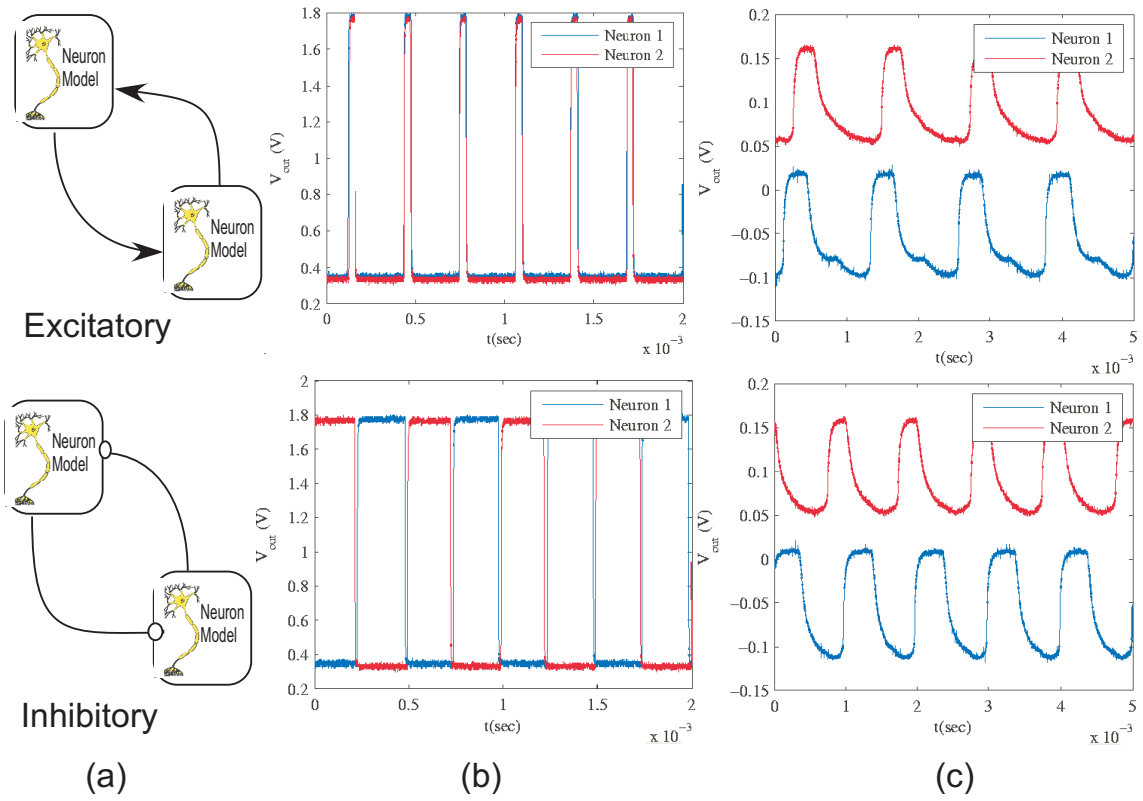


Figure 42: **Coupled Oscillators:** (a) Two neurons coupled by excitatory and inhibitory connections. (b) Measured output from integrate and fire neurons coupled with excitatory (top) and inhibitory (bottom) synapses. (c) Measured output from Hopf neurons coupled with excitatory (top) and inhibitory (bottom) synapses. Absolute voltage values not shown.

the neuron was biased so that input from any one dendrite did not evoke an action potential. Since the neuron is of the Hopf type, it has a resonant frequency, the inverse of which we can call a resonant time period. Input signals that arrive at the soma at time intervals separated by the resonant time period and its multiples have higher chances of evoking action potentials since their effects add in phase.

Figure 44(b) shows the pattern of inputs applied. Cases 1 to 3 shows three instances of input pulses with increasing time difference, t_d, between them. We show the case when the three pulses are on the same dendrite but the same experiment has been done with input pulses on different dendrites too. Figure 44(c) plots the resulting membrane potential for different values of t_d. For case 1, the small value of t_d leads to aggregation of the EPSP signals making the neuron fire an action potential. This behavior is similar to a coincidence detector. When t_d is very large as

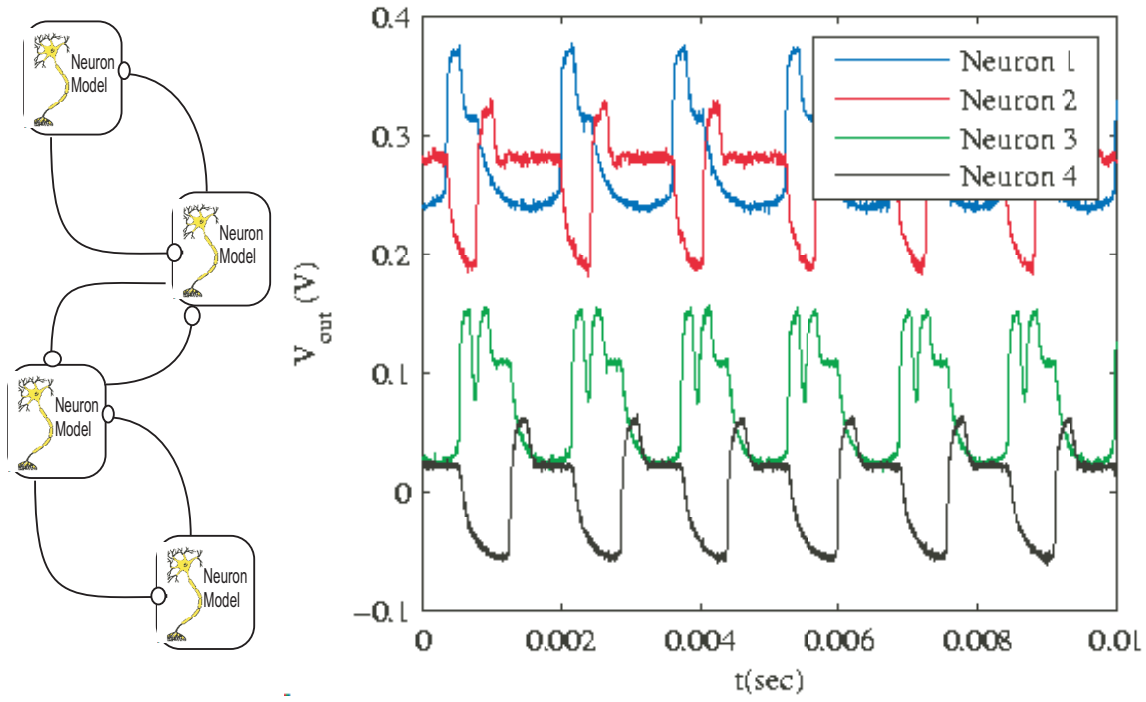


Figure 43: **Central Pattern Generator:** (a) A set of four neurons coupled to its nearest neighbors with inhibitory connections. This models the central pattern generator in many organisms [4, 113]. (b) Measured waveforms of the four Hopf neurons showing different phases of oscillations. Absolute voltages are not shown.

in case 3, the EPSP signals are almost independent of each other and do not result in a spike. However, at an intermediate value of the time difference, we do observe multiple spikes because of in-phase addition of the EPSP (case 2). The reason for this behavior is the value of t_d in this case is close to the resonant time of the Hopf neuron as mentioned earlier. The lengths of the dendrite segments can be modified such that the neuron spikes only when the inputs on the different branches are separated by specific time delays. This serves as one example of possible dendritic computation.

5.6 Discussions

Having described several circuits and systems that can be implemented on the chip in earlier sections, we now discuss a few aspects relating to the computational efficiency, accuracy and scaling of this reconfigurable approach. We also talk about obtaining parameters for the Hopf neurons from Hodgkin-Huxley type parameters.

5.6.1 Computational Efficiency

The efficacy of the analog implementation can be appreciated by considering the effective number of computations it is performing. Let us consider the case of the central pattern generator presented in the last section. In this case, we can model the whole system by a set of differential equations and compute the number of multiply-accumulate (MAC) operations needed to perform the same computation on a computer. We consider an RK 4th order integrator (neglecting possible numerical problems because of multiple time scales) with a time step of 20 μsec (since the spiking activity is on a scale of msec). There are 5 function evaluations per integration step with around 40 MAC needed for every function evaluation (cosh, exp etc). There are at least 12 state variables in this system (2 per neuron and 1 per synapse dynamics block) leading to a computational complexity of 120 MMAC/s. Power consumption for this computation on a 16 bit TI DSP is around 30 mW (excluding power dissipation for memory access) [75]. Our analog implementation consumes 0.74 μW resulting in a performance of 162 GOPS/mW. The area needed for this system was .024 mm^2 in addition to routing. However, using silicon area as a metric is misleading since in this case a lot of the area is traded for achieving reconfigurability. Compared to a DSP, single instruction multiple data (SIMD) paradigm based cellular non-linear network (CNN) systems [36, 39, 62, 75] report performances that are closer to this chip. Though these systems do not replicate biological behavior at the level of ion-channels like our chip, their designs are nevertheless based on abstractions of neural systems. The reported performance for some of these chips are 1.56 GOPS/mW [39] and 0.08 GOPS/mW [75], significantly less than our chip. There are, of course, other functionalities that these chips can do better than ours. It should also be noted, that the DSP performs 16 bit computations, while the analog one is less accurate. These inaccuracies are described next.

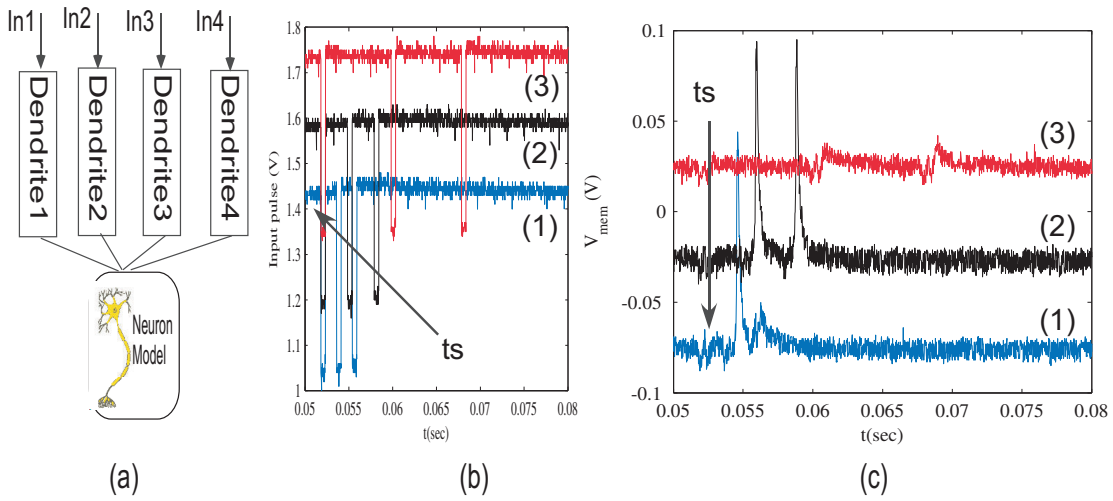


Figure 44: **Coincidence detector:** (a) Schematic of a neuron with four different inputs incident on four dendritic branches. (b) This figure is indicative of the timing relationships between the input pulses (voltages are shifted for better viewing). t_s indicates the time when the first pulse was applied. (c) When the time delay between inputs is small, we see the classical aggregation of EPSP leading to a spike in case 1 while there is no spike in case 3 because of the large time delay between input pulses. Case 2 shows multiple spikes since the Hopf neuron is most excitable when the inter-pulse interval is close to the resonant time of the neuron. Absolute voltage are not shown here.

5.6.2 Sources of Error

The most obvious source of error is finite resolution in setting the circuit parameters and synaptic weights. This relates to FG programming accuracy, which, in our case is limited by the resolution of the ADC for measurements. Our measurement approach creates a floating-point ADC [11] with an effective resolution of around 9.5 bits.

The next source of error stems from mimicking a biological phenomenon by silicon circuits. Some of the approaches we presented are based on qualitative similarities between the silicon circuit and its biological counterpart. For example, Fig. 39 depicts the mismatch between a synaptic EPSP and a biological one is around 10% corresponding to around 3.5 bits. In general, this error is difficult to analyze and depends on the desired computation.

Finally, thermal noise presents a fundamental limit to the computational accuracy. The low-current, low-capacitance designs we presented save on power in exchange for thermal noise. This is actually close to the computational paradigm employed by

biology and hence is not necessarily a problem.

5.6.3 Scaling

To expand these silicon systems to mimic actual biology, multiple chips need to be interconnected. The modular architecture of our chip does allow tiling of several chips. In that case, however, all-to-all connectivity has to be sacrificed due to the limited number of routing lines. This is also not unlike biology where local interconnects are more dense than global connections. To allow more flexibility in inter-chip connections, the next generation of these chips are being designed with address-event (AER) support [19].

5.6.4 Parameter Conversion

To use this analog chip in place of digital simulations, there needs to be a method to bias the analog neurons to perform computations similar to the digital simulation. We will show parameter conversion for a H-H type potassium channel and transient sodium channel [47] to silicon channel models as described in [37]. An obvious question that arises is "Should we model the biological channel or should we model a set of equations that model biology ?" However, since many neuroscientists already use the H-H model, this kind of an automated conversion was deemed useful for porting their already existing designs. In spirit, this seems similar to extracting EKV model parameters from BSIM parameters, both of which model a transistor's operation, but the EKV is closer to the physics of the device while the BSIM model is more of a curve fit.

The model put forward by Hodgkin and Huxley described ion channels as time-variant non-linear conductances controlled by activation or inactivation of certain gating molecules. In the silicon implementation, transistors operating in subthreshold regime with gating amplifiers are used instead to model channel behavior. Without going into a debate over which model is more biologically accurate, we shall try to find

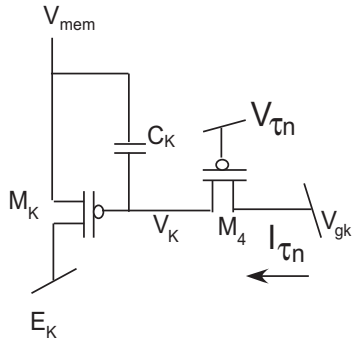


Figure 45: **Potassium channel:** A silicon implementation of a potassium channel as described in [37].

parameters for both representations such that their behaviors are similar in a small range of membrane potentials. The operation of the model near resting potential is probably the most important to determine the spike timing and hence we equate the behavior of both models close to rest. Our approach mainly consists of equating DC operating points and small signal transconductances of both models.

We shall begin by discussing how to choose parameters for the potassium channel.

5.6.4.1 Potassium Channel

Figure 45 shows the schematic for a silicon potassium channel.

The equation for the potassium channel population described in Hodgkin-Huxley's original paper [47] is:

$$\begin{aligned} I_k &= g_k \times n^4 \times (V_{mem} - E_k) \\ \frac{dn}{dt} &= \frac{n_{ss}(V_{mem}) - n}{\tau_n} \\ n_{ss}(V_{mem}) &= \frac{1}{1 + e^{\frac{V_{n,half} - V_{mem}}{n_k}}} \end{aligned} \tag{22}$$

where I_k is the potassium channel current, g_k is the conductance of the channel population for a certain cross sectional area, V_{mem} is the membrane potential, E_k is the reversal potential for potassium ions, n is the activation variable describing the fraction of potassium channels that are open at a particular instant, n_{ss} describes the

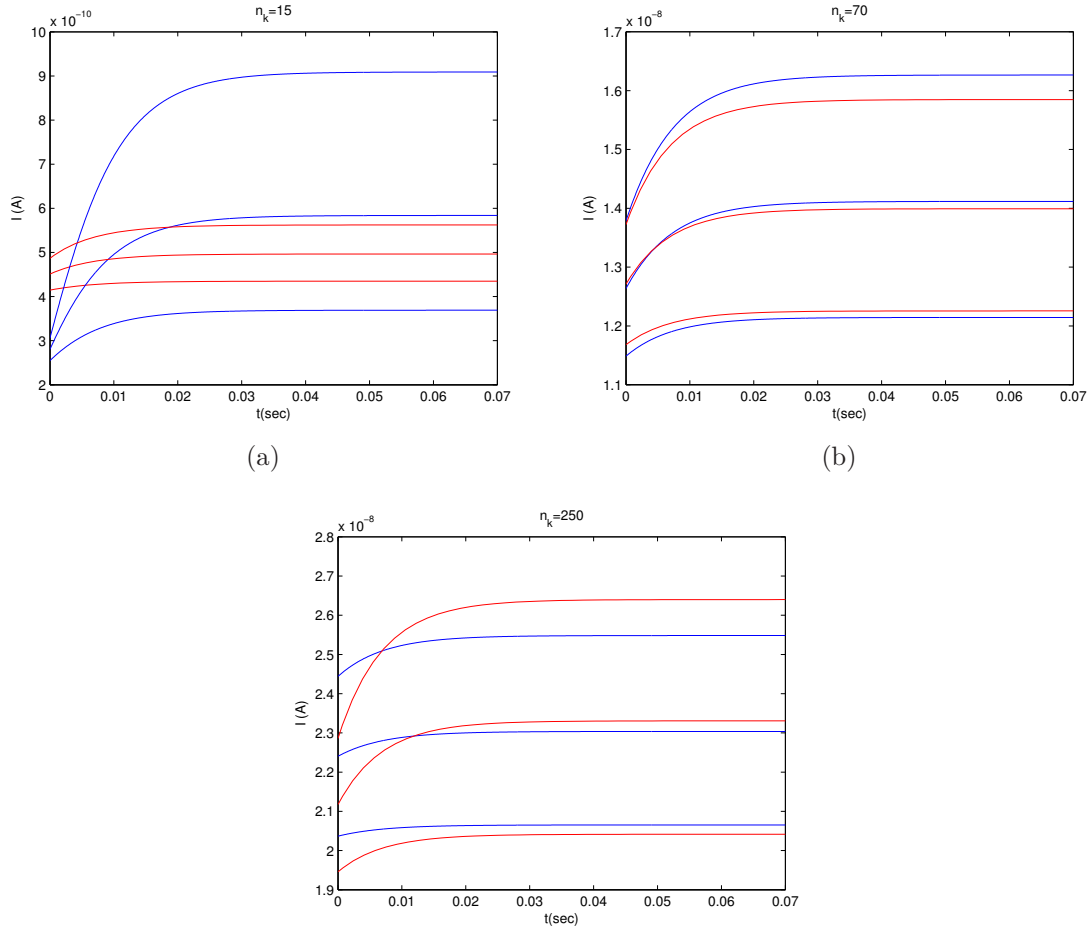


Figure 46: **Potassium channel voltage clamp:** Voltage clamp responses for H-H channels (blue) and silicon implementation (red) are plotted for increasing steps of 1.67 mV each. The three plots depict voltage clamp data for three different slopes (n_k) of the H-H activation function with the values being 15, 70 and 250 mV^{-1} respectively.

steady state value of the potassium activation which has a sigmoidal dependence on the membrane potential and τ_n is the time constant for the activation of potassium. Typically, H-H models have a voltage dependent time constant. But for our analysis, we shall ignore the voltage dependence and consider an average value of the time constant for membrane voltages in the vicinity of the resting potential. The

corresponding equation for the silicon version of the potassium channel is:

$$I_k = I_{k0} e^{-\frac{\kappa v_k}{U_T}} (e^{\frac{V_{mem}-E_k}{U_T}} - 1)$$

$$v_k = V_k - V_{gk}$$

$$\frac{dv_k}{dt} = -\frac{I_{\tau n}}{C_k}(e^{\frac{v_k}{U_T}} - 1) \tag{23}$$

where $I_{k0} = I_0 e^{-\frac{E_k - \kappa V_{gk}}{U_T}}$, $I_{\tau n} = I_0 e^{\frac{V_{gk} - \kappa V_{\tau n}}{U_T}}$, κ is the coupling from the transistor's gate to channel and U_T is the thermal voltage. The two voltage biases to be chosen for the silicon channel are V_{gk} and $V_{\tau n}$. These two biases are represented in the mathematical model by the parameters I_{k0} and $I_{\tau n}$ respectively. $I_{\tau n}$ can be obtained by equating τ_n with the linearized time constant for 23 to get:

$$I_{\tau n} = \frac{C_k U_T}{\tau_n} \tag{24}$$

I_{k0} can be obtained by equating the steady state currents from both the channels at some particular voltage. A possible choice of this voltage might be the resting potential, V_{rest}. Using that choice, we get:

$$I_{k0} = \frac{g_k \times n_{ss}^4(V_{rest}) \times (V_{rest} - E_k)}{e^{\frac{V_{rest}-E_k}{U_T}} - 1} \tag{25}$$

With these approximations, we can plot the current resulting from both the H-H and the silicon channel for voltage clamps. Figure 46 plots the resulting voltage clamp from both channel models for different values of n_k, the slope of the activation function. Results from the H-H model are plotted in blue while the currents from the silicon implementation are plotted in red. The voltage clamp steps start from 15 mV above E_k and increase in values of 1.67 mV. It can be seen that there is a good agreement for the currents for $n_k = 70$ while the quality of the fit degrades for other values. The reason for this can be attributed to the absence of any gating amplifier controlling the gate of M_k, the potassium channel transistor. Hence, a better fit may be obtained by modifying this silicon channel.

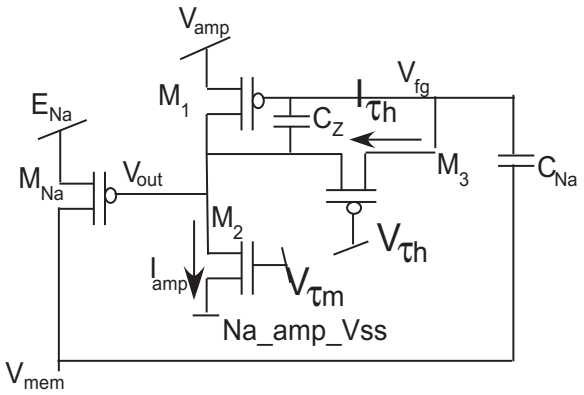


Figure 47: **Sodium channel:** A silicon implementation of a sodium channel as described in [37].

5.6.4.2 Sodium Channel

Figure 47 shows the schematic for a silicon sodium channel.

The transient sodium channel in the Hodgkin Huxley model is more complicated than the potassium since it also has an inactivation variable along with the activation variable. The equation for it is given by:

$$\begin{aligned} I_{na} &= g_{na} \times m^3 h \times (V_{mem} - E_{na}) \\ \frac{dm}{dt} &= \frac{m_{ss}(V_{mem}) - m}{\tau_m} \\ \frac{dh}{dt} &= \frac{h_{ss}(V_{mem}) - h}{\tau_h} \\ m_{ss}(V_{mem}) &= \frac{1}{1 + e^{\frac{V_{m,half} - V_{mem}}{m_k}}} \\ h_{ss}(V_{mem}) &= \frac{1}{1 + e^{\frac{V_{h,half} - V_{mem}}{h_k}}} \end{aligned} \tag{26}$$

where m is the activation variable describing the fraction of sodium channels that are open at a particular instant, h is the inactivation variable describing the fraction of activated gates that are inactivating, m_{ss} and h_{ss} describe the steady state value of the sodium activation and inactivation, τ_m and τ_h are the time constants for activation and inactivation of sodium and all other parameters are sodium channel equivalents for similarly named potassium channel parameters. Similar to the potassium channel, we again accurately model the behavior of the channel close to the resting potential

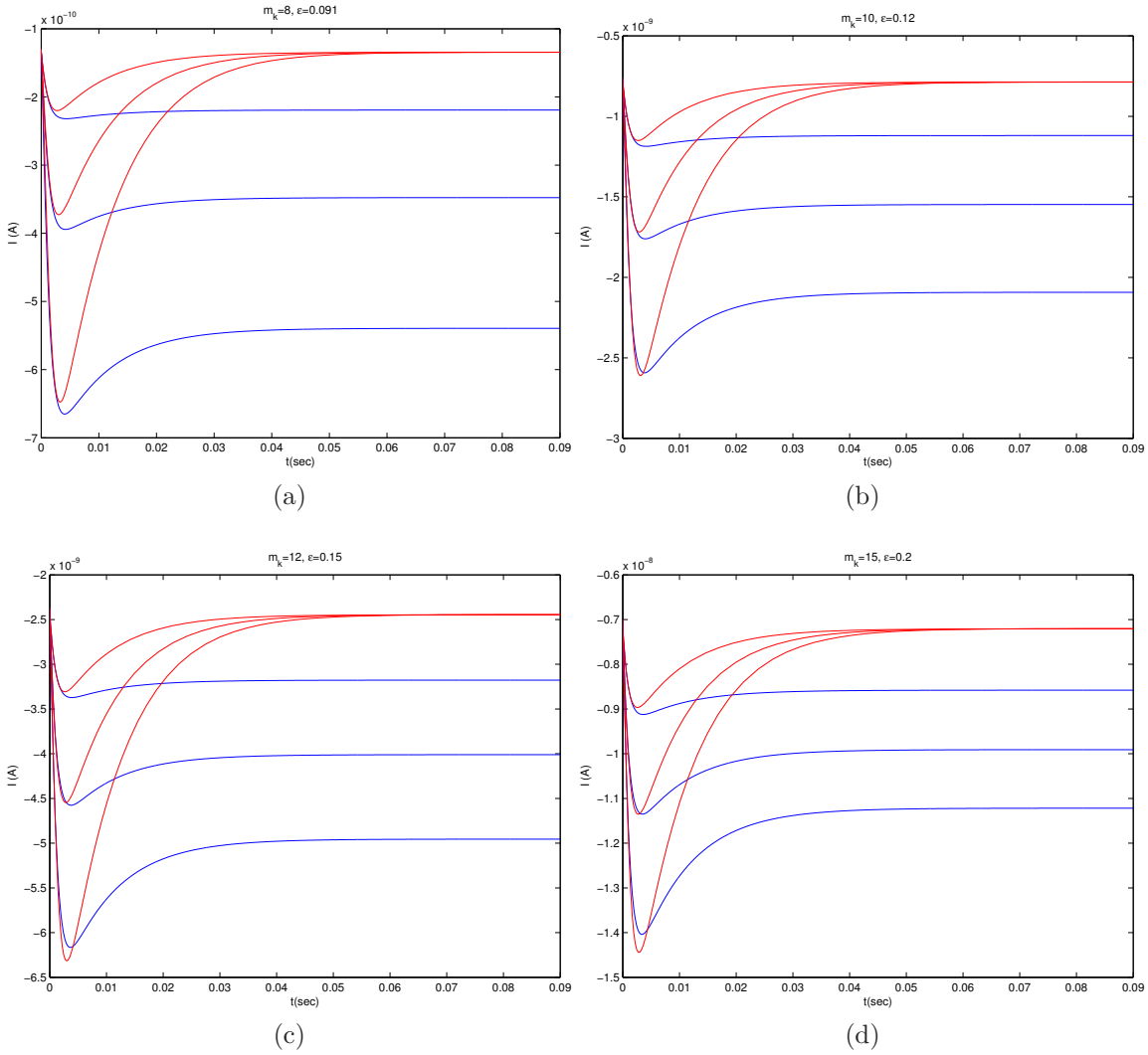


Figure 48: **Sodium channel voltage clamp:** Voltage clamp responses for H-H channels (blue) and silicon implementation (red) are plotted for increasing steps of 1.67 mV each. The four plots depict voltage clamp data for four different slopes (m_k) of the H-H activation function with the values being 8, 10, 12 and 15 mV^{-1} respectively.

due to the importance of the subthreshold dynamics in determining spike timing.

For the silicon implementation, a bandpass amplifier is used to mimic the effect of fast activation followed by slow inactivation. The corresponding equations are:

$$
\begin{aligned}
I_{na} &= I_{na0} e^{-\frac{\kappa \Delta V_{out}}{U_T}} \\
\Delta \dot{V_{fg}} &= \frac{I_{amp}}{C_{na}} e^{-\frac{\kappa \Delta V_{fg}}{U_T}} - \frac{I_{amp}}{C_{na}} (1 - e^{\frac{-\beta U_T - \Delta V_{out}}{U_T}}) \\
\Delta \dot{V_{out}} &= (1 + 1/\epsilon) \Delta \dot{V_{fg}} + \frac{I_{\tau h}}{C_{na} \epsilon} (e^{\frac{\Delta V_{fg}}{U_T}} - e^{\frac{\Delta V_{out}}{U_T}})
\end{aligned}
\tag{27}
$$

where I_{na0} is the steady state current in the sodium channel, $\epsilon = C_z/C_{na}$, $V_{out0} - Na\_amp\_vss = \beta U_T$ where V_{out0} is the steady state value of V_{out} and ΔV represents deviations of a voltage variable from its steady state value. I_{na0} can be obtained by equating the rest currents to get:

$$I_{na0} = g_{na} \times m_{ss}^3(V_{rest}) \times h_{ss}(V_{rest}) \times (V_{rest} - E_{na}) \tag{28}$$

where the ohmic nature of M_{na} has been neglected since V_{rest} is typically lesser than E_{na} by more than $4U_T$.

The mid frequency voltage gain of the bandpass sodium amplifier is approximately $1/\epsilon$, the ratio of C_{na} and C_z. This gain is determined by the transconductance gain of the sodium channel, one factor in determining which is the slope, m_k of its activation curve m_{ss}. Let us elaborate this point a bit more. Since the sodium channel creates transient currents, the gain of the channel can be estimated by measuring the peak current values during a voltage clamp experiment. Then the transconductance gain of the channel can be defined as the ratio of the peak current value to the magnitude of the small step applied. To estimate the peak current we take advantage of the separation of time scales of activation and inactivation and assume that the channel achieves its peak current when the activation variable reaches its steady state. In other words, we assume that by the time the activation variable reaches its steady state, the inactivation variable has not changed appreciably from its initial value before the step. With these approximations, the transconductance gain, $\overline{G}$ of a H-H type transient channel can be derived as:

$$\begin{aligned} I_x &= g_x \times m^\alpha(V) \times h(V) \times (V - E_x) \\ \overline{G_{hh,x}} &= g_x \times m_{ss}^\alpha(V_{rest}) \times h_{ss}(V_{rest}) + \frac{g_x \alpha}{x_k} \times m_{ss}^\alpha(V_{rest}) \times (1 - m_{ss}(V_{rest})) \times h_{ss}(V_{rest}) \\ &\times (V_{rest} - E_x) \end{aligned} \tag{29}$$

where parameters have similar definitions to the ones described for the sodium channel

which had subscripts of 'na' instead. Now, the parameter ϵ can be estimated as:

$$\epsilon = \frac{Gm, M_{na}}{\overline{G_{hh,na}}} = \frac{\kappa I_{na0}}{U_T \overline{G_{hh,na}}} \tag{30}$$

where Gm, M_{na} is the small signal transconductance of the sodium channel transistor.

The voltage biases $V_{\tau m}$ and $V_{\tau h}$ that set the bias currents I_{amp} and $I_{\tau h}$ can be obtained by equating the linearized activation and inactivation time constants from 27 with the H-H time constants to get:

$$\begin{aligned} I_{amp} &= \frac{C_{na} U_T}{\kappa \tau_m} \\ I_{\tau h} &= \frac{\epsilon C_{na} U_T}{\tau_h} \end{aligned} \tag{31}$$

Finally, the value of β can be estimated by calculating the maximum possible current in the sodium channel. This value can be roughly approximated as follows:

$$\begin{aligned} I_{na,max} &= g_{na} \times m_{ss}(V_{max})^3 \times h_{ss}(V_{rest})(V_{max} - E_{na}) \\ \beta &= \frac{\ln\left(\frac{I_{na,max}}{I_{na0}}\right)}{\kappa} \end{aligned} \tag{32}$$

where V_{max} is that value of membrane voltage for which the expression of $I_{na,max}$ attains its maximum. This value can be found iteratively with a good starting guess being a value V_0 such that $m_{ss}(V_0) > 0.9$.

Estimating the parameters for the silicon channel in this way, let us compare the voltage clamp responses for both the models. Figure 48 plots the voltage clamps of both channels with the voltage steps being same as the potassium channel experiment. Like the earlier case, results from the H-H model are plotted in blue while the currents from the silicon implementation are plotted in red. Again, four different cases are plotted for different values of the sodium activation slope, m_k. Unlike the earlier case, the gain of the sodium amplifier is modified for these different cases leading to a good approximation of the peak current values. However, it should be noted that in a silicon implementation, the option of changing capacitor ratios is not available.

Hence, in a real implementation, the equivalence will be good for a certain value of m_k and degrade otherwise. Also, it should be noted that the final steady state currents from the silicon channel do not match the H-H channel values. The reason for this is the silicon model of the sodium channel assumes almost complete inactivation at the end of the step. But for the parameters of h_{ss} chosen in this simulation, h_{ss} is not small enough when V is close to the value for V_{rest}. Hence, the final values of the currents for the H-H channel simulation do not reduce to be close to the rest current.

5.7 Conclusion

We presented a reconfigurable integrated circuit for accurately describing neural dynamics and computations. There have been several earlier implementations of silicon neural networks with a dense synaptic interconnect matrix. But all of them suffer from one or more of the following problems: fixed connectivity of the synaptic matrix [26], inability to independently control the neuron parameters since they are set globally [51, 112] and excessively simple transfer-function based neuron models [107].

In the chip we present, both the topology of the networks as well as the parameters of the individual blocks can be modified using floating-gate transistors. Neuron models of complexity varying from integrate and fire to Hodgkin-Huxley can be implemented. Computational area efficiency is considerably improved by implementing synaptic weight on the analog switch matrix resulting in all to all connectivity of neurons. We demonstrate dynamics of integrate and fire neurons, Hopf neurons of Hodgkin-Huxley type, inhibitory and excitatory synapses, dendritic cables and central pattern generators. In the next section, I describe a different chip employing the same architecture but with different CAB components suited for signal processing. This serves as an example for the generality of the architecture and underlines the possibility of using the FPAA platform for different applications.

CHAPTER VI

RECONFIGURABLE ANALOG SIGNAL PROCESSOR

This chapter describes another FPAA chip employing the same architecture described in 4. The difference of this chip compared to the one presented in the last chapter is that the CAB components are suited for analog signal processing tasks. These components are also programmable, with this flexibility being provided by floating-gate elements. In the following sections, we first describe the CAB components in this chip and then present several small circuits demonstrating the operation of the CAB elements.

6.1 CAB Elements: Characterization and Simple Circuits

6.1.1 Cab Elements

The CABs are of two major types as shown in Fig. 49(a). Figure 49(b) depicts the components in the two CABs. The first one has three operational transconductance amplifiers(OTA), three floating capacitors (500 fF each), two multi-input floating gates which can be used for constructing translinear circuits using MITE architectures, a voltage buffer, a transmission-gate with dummy switch for switched-capacitor applications, and NMOS/PMOS transistor arrays with two common terminal for easily constructing source-follower or current-mirror topologies. All the OTAs are biased using floating-gate transistors giving the user the option to tradeoff bandwidth, noise and power. Cascode biasing circuits based on [72] are included as well. Two of the OTAs have floating-gate differential pairs (FGOTA) which enable programming the offset of the amplifier as well as provide wide input linear-range that is essential to reduce distortion in Gm-C filters and oscillators. The second type of CAB has two folded Gilbert multipliers and a floating-gate current mirror in addition to a wide

linear range OTA. The multiplier also has floating-gate differential pairs to reduce distortion. The FG current mirror can be used to convert the differential output currents of the multiplier into a single ended voltage output. These CAB components can be connected using the switch-matrix consisting of floating-gate switches, which unlike other digital switch implementations, can also be used for analog computations.

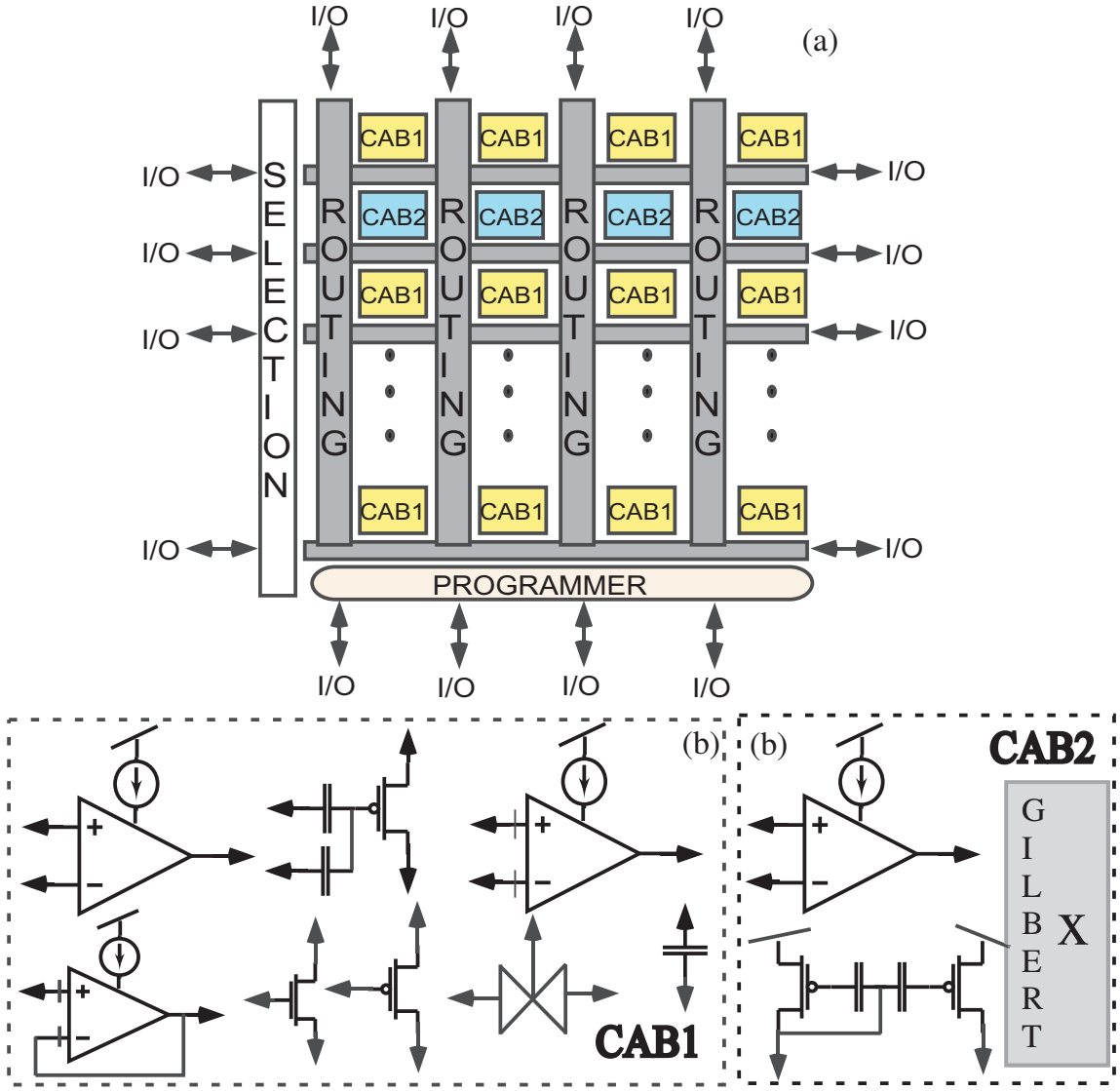


Figure 49: **Switch matrix:** (a) The architecture of the chip shows two different CABs. (b) The components in the two types of CABs vary from programmable transconductors and multipliers to transistors and capacitors.

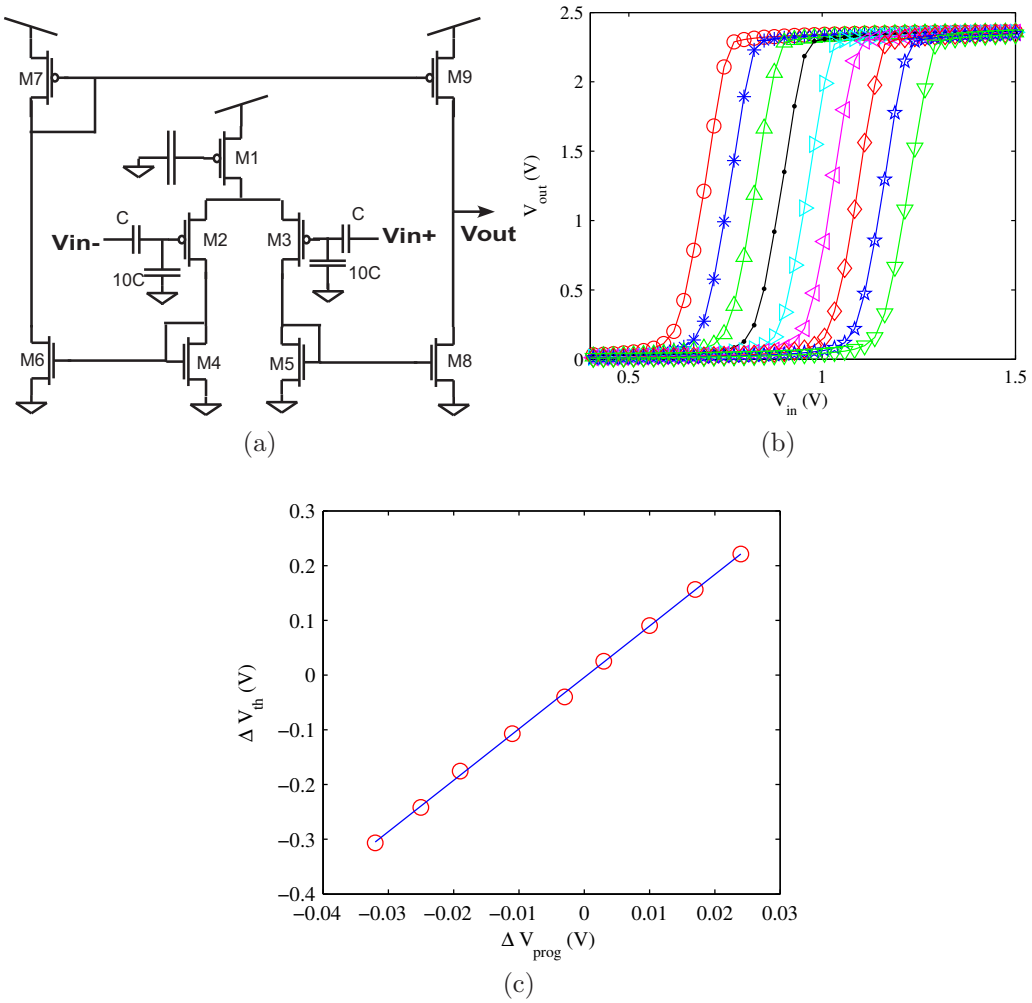


Figure 50: **Variable offset comparator:** (a) Schematic of a floating-gate input OTA whose input offset can be programmed using the FG inputs. It is used as a comparator in this experiment. (b) Different values of programmed charge difference on the differential inputs leads to different measured trip points for the comparator. (c) Relation between the comparator trip points and the programmed charge difference shows an average deviation of 5.8 % from linearity.

6.1.2 Floating-gate input Operational Transconductance Amplifier (FGOTA)

One of the most versatile circuit elements on the chip is the FGOTA [23]. The circuit is shown in Fig. 50(a). It consists of a differential amplifier followed by a push-pull output stage. The bias current of the amplifier can be programmed using an FG device (M1). The differential pair transistors (M2 and M3) are also floating gates which allows the user to program a desired input offset voltage. The input is applied to the differential pair through a capacitive divider that has a 1:9 ratio. This allows

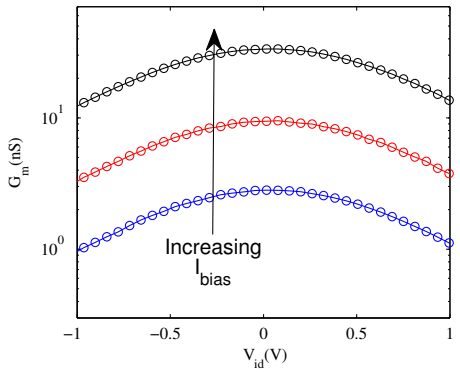


Figure 51: **Wide input linear range:** A capacitive divider is placed before the input of a traditional differential pair to provide an effectively wider linear range. This circuit has a measured linear range of 673 mV for 5 % degradation in transconductance.

a wider input linear range by effectively degenerating the transconductance. The advantage of this method over other degeneration schemes is that the capacitors add virtually no noise. Mismatch in the series capacitors (of nominal value C), however, directly degrades the CMRR. In future versions, cascode transistors will be used in the output stage to improve the output impedance and DC gain.

Since arbitrary input offsets (limited by the precision of programming charge) can be stored in the input differential pair, the FGOTA can be used as a comparator for a flash ADC and eliminates the need for a resistor ladder. Figure 50(b) shows a case where 9 different offset values are set by programming the charge difference. Figure 50(c) shows the relation between the programmed difference in floating-gate voltage (as extrapolated from the current measurements during programming) and the measured trip-point of the comparator. The average deviation from linearity is 5.8 % and can be attributed to errors in programming, extrapolating the floating-gate potential and measuring the trip point from the high gain curve. The power dissipation of the circuit is 9.6 μW and can be reduced if smaller bandwidth is needed. Since there are sixty FGOTAs in the chip, a 5 bit flash ADC (without the thermometer decoder) can be compiled.

The widening of the input linear range is depicted in Fig. 51 by plotting the transconductance of the block with respect to input differential voltage. Assuming

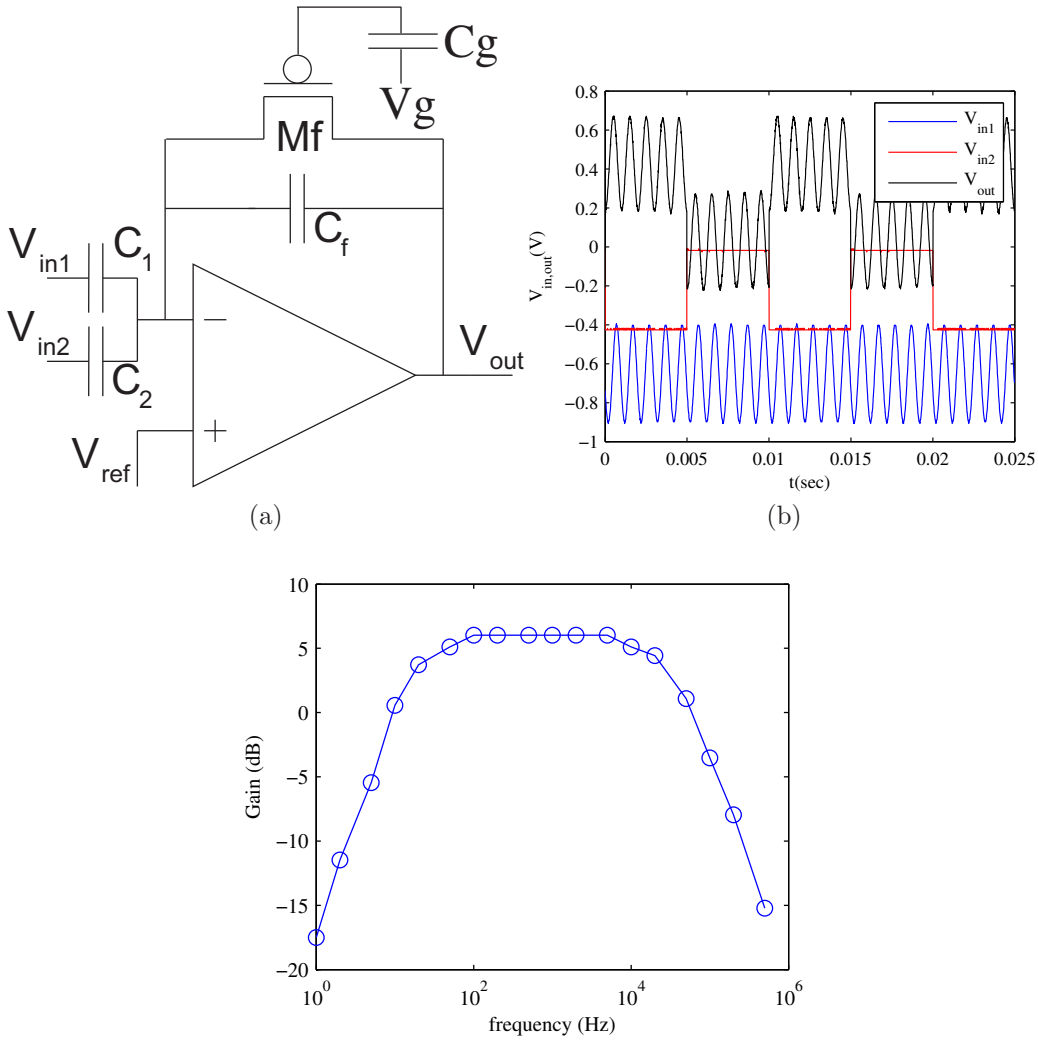


Figure 52: **AC Summing circuit:** (a) Schematic of a capacitive summing circuit. (b) The summed output when a square wave and a sinusoid are applied to the inputs. (c) The circuit has a bandpass response with the lower and higher poles being set by the conductance of Mf and the transconductance of the OTA. Measured corner frequencies are 20 Hz and 20 kHz.

subthreshold operation, the I-V relation is:

$$I_{out} = I_{bias} \tanh(\frac{\kappa\alpha(V_{in+} - V_{in-})}{2U_T}) \tag{33}$$

where α is the attenuation factor due to the divider, I_{bias} is the bias current of the stage, κ is the coupling of the gate to the channel of a subthreshold MOSFET and U_T is the thermal voltage. Figure 51 plots the transconductance for three bias currents. The linear range is 673 mV differential for 5 % degradation in G_m.

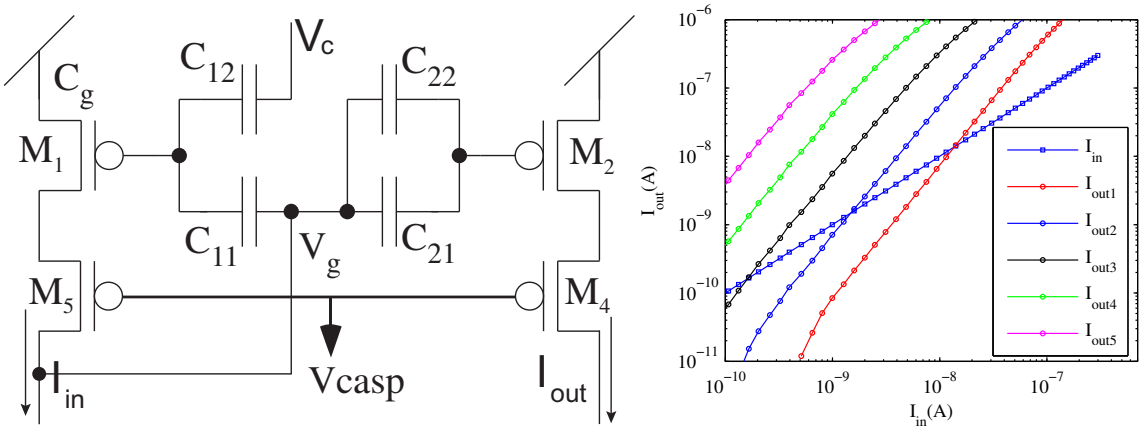


Figure 53: **Squaring circuit:** (a) Circuit schematic for a MITE based implementation of a squaring circuit. (b) Measured data shows an exponent of 1.95 with the deviation being probably due to mismatch between the MITE transistors and the capacitors.

6.1.3 Capacitive Summer

Figure 52(a) shows the circuit for summing the two signals V_{in1} and V_{in2}. C_1, C_2 and C_f are capacitors of value 500 fF each. The transconductance amplifier is similar to the FGOTA described in the earlier subsection but without a floating-gate input differential pair. Mf is a FG device from the switch matrix (operating in subthreshold) that acts like a small conductance, g_f, with its value given by:

$$g_f = \frac{I_0 e^{\frac{V_{ref} - \kappa V_{fg}}{U_T}}}{U_T} \tag{34}$$

where V_{fg} is the potential of the floating node, I_0 is the pre-exponential factor in the current equation of a subthreshold PMOS and κ was mentioned earlier in (33). This circuit has a bandpass response [79] with the lower and higher poles, p_1 and p_2 given by:

$$\begin{aligned} p_1 &= \frac{g_f}{C_f} \\ p_2 &= \frac{G_m}{C_{eff}} \end{aligned} \tag{35}$$

where G_m is the transconductance of the OTA and $C_{eff} = (C_o C_T - C_f^2)/C_f$, $C_o = C_L + C_f$ and $C_T = C_1 + C_2 + C_f + C_w$. C_w and C_L are parasitic capacitances at the

inverting input and output of the OTA respectively and are not shown in the figure. In the pass-band the output voltage is given by:

$$V_{out} = -\frac{C_1}{C_f}V_{in1} - \frac{C_2}{C_f}V_{in2} \tag{36}$$

which is a weighted summation of the two inputs. Figure 52(b) shows the output summed voltage when the inputs are a square wave and a sinusoid. Figure 52(c) shows the measured frequency response of the circuit when the same sinusoid is applied to both inputs. G_m and g_f were programmed to get lower and higher cut-off frequencies at 20 Hz and 20 kHz respectively. The nominal pass-band gain is 6 dB with a maximum error of 0.021 dB.

6.1.4 Current-mode Translinear circuits

Translinear circuits introduced by Gilbert are very popular for their versatility in implementing signal processing functions. In particular, multi-input translinear elements (MITE) [73] have been shown to successfully synthesize various static and dynamic functions. This FPAA chip has a couple of two input MITE elements in every CAB of type 1. Figure 53(a) shows the schematic of a simple circuit to implement the static function of squaring the input. All the capacitors are nominally equal to 110 fF. The output current is measured using an off-chip pico-ammeter. The output current is given by:

$$\begin{aligned} I_{out} &= I_{ref2}\left(\frac{I_{in}}{I_{ref1}}\right)^{1+a/b} \\ a &= -\frac{\kappa_2(C_{21}+C_{22})}{C_{T2}U_T} + \frac{\kappa_1(C_{11}+C_{gd1})}{C_{T1}U_T} \\ b &= -\frac{\kappa_1(C_{11}+C_{gd1})}{C_{T1}U_T} \\ C_{Tj} &= C_{j1}+C_{j2}+C_{gj}, j = 1,2 \end{aligned} \tag{37}$$

where quantities with subscripts 1 or 2 refer to parameters for M1 or M2. I_{ref1}/I_{ref2} are constants associated with M1/M2 and are controlled by their respective floating-gate charges. I_{ref1} is also controlled by V_c. For nominally matched parameters of

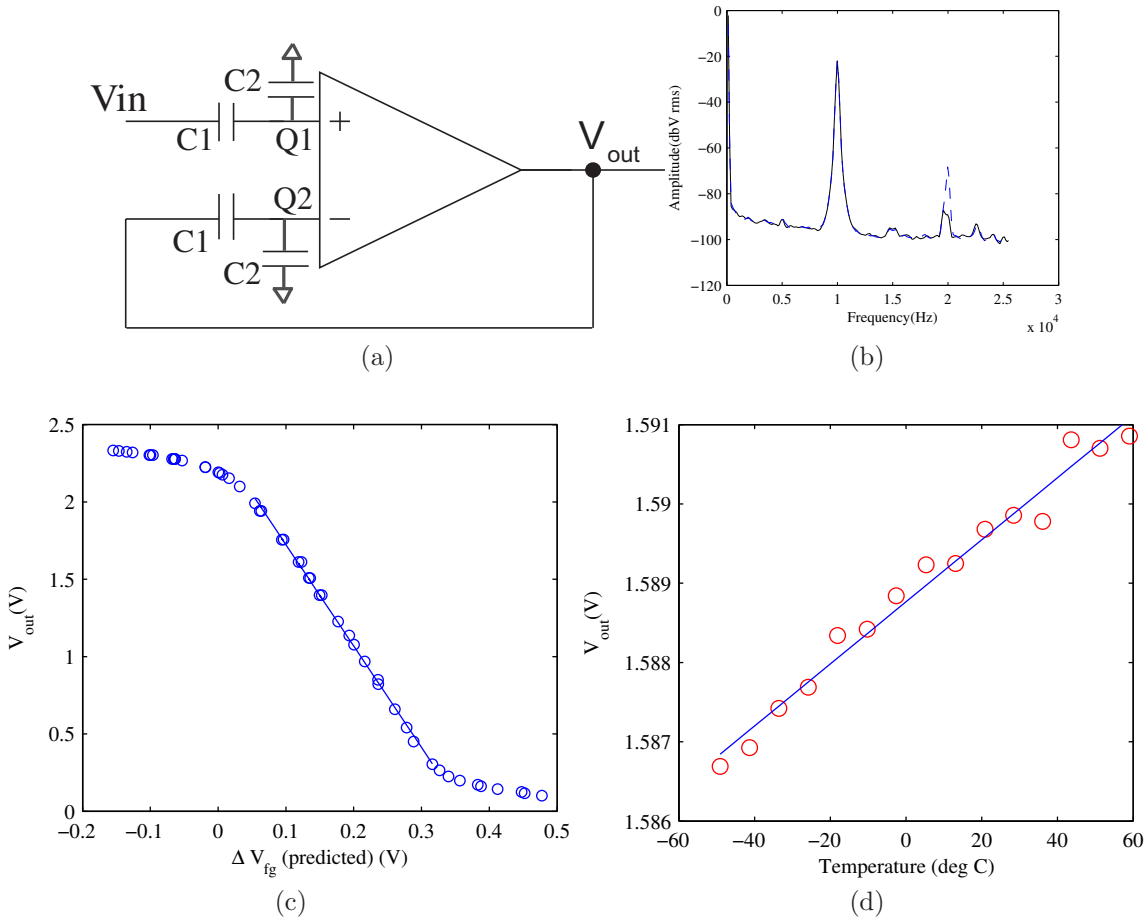


Figure 54: **FGOTA buffer:** (a) Circuit schematic of the floating-gate input buffer. (b) Offset voltage reduction leads to a reduction in 2nd harmonic at the output of the buffer. Measurements show an improvement of 19 dB.(c) The programmable input offset voltage leads to a level shift in the output. This can be used as a reference voltage generator with V_{in} connected to V_{dd}. Measured reference voltage is linearly related to programmed offset. (d) Since the floating-gate charge does not change appreciably with temperature, the reference has a low TC. Measured TC is equal to 24.6 ppm.

both devices and ignoring C_{gd1}, 'a' and 'b' are equal leading to a squaring circuit.

Figure 53(b) plots the output current against the input current for several values of V_c. The average slope of the plots is 1.95 with the error resulting from mismatch. Many other static and dynamic functions can be synthesized limited only to the number of MITE devices in the chip.

6.1.5 Programmable Offset Buffer

An FGOTA circuit configured as a voltage buffer has been placed in the CAB due to its frequent need. The circuit is shown in Fig. 54(a). The ratio of C2 and C1 is 8 as mentioned earlier. Since an offset can be programmed by setting a desired charge difference on the floating nodes, this can serve as a level-shifting buffer with the level shift being approximately 9 times the programmed offset voltage. The offset can also be reduced to zero (within programming precision) which leads to a reduction in the second harmonic signal at the output. Figure 54(b) plots the output spectrum of the buffer with a 10 kHz sinusoid at its input. The output is plotted without and with offset correction. The spurious free dynamic range (SFDR) is 65.2 dB after correction, 19 dB better than earlier.

Because of the level shifting properties mentioned earlier, the output voltage can be set to an arbitrary value within a range even if the input is fixed to V_{dd}. This method is used to create a reference voltage generator. The capacitive divider ensures that generating a wide range of reference voltages does not need a wide input common mode range (ICMR) for the amplifier. Figure 54(c) plots the output reference voltage against the programmed offset voltage (predicted from current measurements of the floating-gates). In this experiment Q1 was fixed to a value while Q2 was varied. The curve saturates near the power rails possibly due to output swing limitations. Within the range of 0.3 V to 1.99 V, the average deviation from linearity is 1.6 %. The power dissipation in this case is 9.6 μW and can be reduced by reducing the bias current.

Since the floating-gate charge does not vary appreciably with temperature [101], this voltage reference also exhibits good temperature stability. Figure 54(d) plots the measured temperature behavior for a particular reference voltage over a range of -50 $^{\circ}C$ to 60 $^{\circ}C$. The measured temperature coefficient (TC) over this range is 24.6 ppm. Assuming perfect matching between the capacitors and no charge loss from the

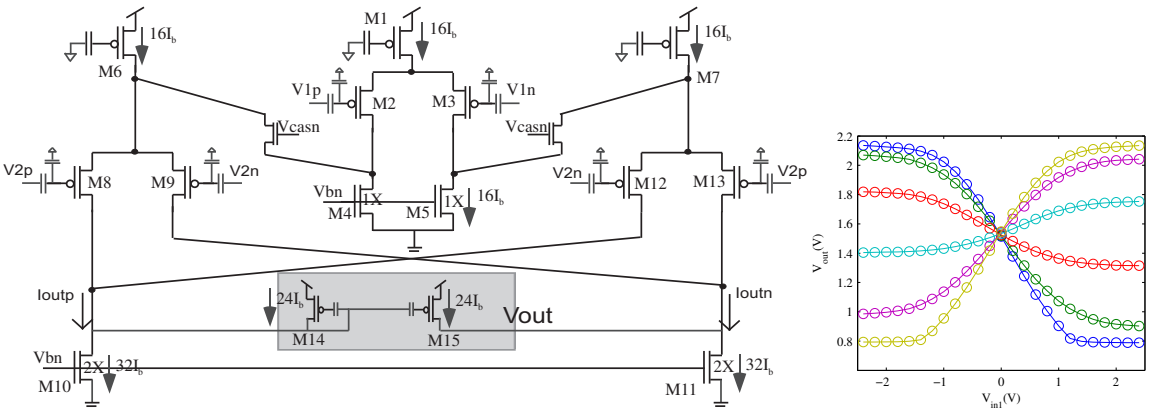


Figure 55: **Gilbert multiplier:** (a) Circuit schematic for a folded Gilbert multiplier. M14 and M15 forms an FG current mirror that can be optionally added to convert the output currents to a single ended voltage. (b) Measured output voltage for sweeping the differential input V_{in1} for different values of V_{in2}. The measured linear range is 2 V differential for a maximum error of 1.5 %.

floating node, the TC of the reference can be shown to be:

$$TC_{Vref} = \frac{1}{Vref}\frac{dVref}{dT} = -\frac{1}{C_1}\frac{dC_1}{dT} = -TC_{C_1} \tag{38}$$

In practice the temperature performance may degrade further because of the variation of the DAC voltage that supplies the voltage at the end of the capacitor C_2 (shown as a small signal ground in Fig. 54(a)) and due to variation of the amplifier's gain. It is shown in [1] that a dual FG design can obtain cancellation of the capacitor TCs because poly-poly capacitors have a voltage dependent TC. Hence, for some values of the reference voltage and floating-gate voltage the TC may be smaller. This option has not yet been explored in our chip.

6.1.6 Folded Gilbert Multiplier

The Gilbert multiplier is a widely used analog component useful for mixers, variable gain amplifiers (VGA), automatic gain control (AGC) and many more applications. However, the stacking of multiple differential pairs leads to voltage headroom issues in a traditional Gilbert cell. This has been solved by folding the signal currents [6,94]. We employ a similar approach as shown in Fig. 55(a). The cascode biases are generated following [72]. In the figure, I_b refers to the current in the bias generation

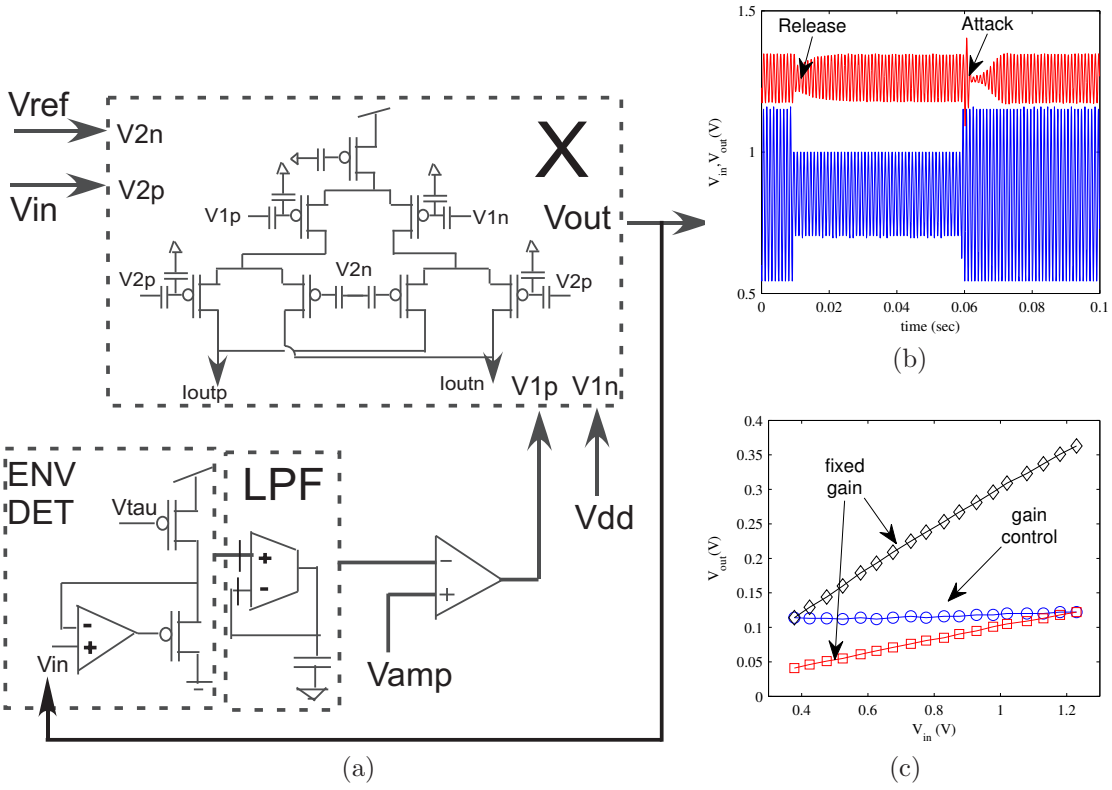


Figure 56: **AGC:** (a) Circuit schematic for an AGC system. (b) Measured transient response of the AGC to large variations of input amplitude. The attack and release times are approximately 11 msec. (c) Measured gain compression curves showing a maximum compression of 28 dB.

circuits. V1 and V2 are the two differential inputs to the multiplier. The multiplier block produces differential output currents which can be converted to a single ended voltage using a FG current mirror (M14 and M15). The low output impedance of M15 (due to C_{gd}) prevents the output from saturating. Figure 55(b) plots the measured output voltage against differential input voltage V_{in1} for several fixed values of V_{in2}. For this case, $I_b = 1$ nA leading to a power dissipation of 230 nW. The maximum deviation from linearity over a differential range of 2 V is 1.5 %.

Next we show an AGC circuit as an example application of the multiplier. The schematic of the AGC is shown in Fig. 56(a). The multiplier is used as a VGA with the gain controlling input being set by feedback. The feedback loop comprises a peak detector, a low pass filter and a high gain amplifier. The desired amplitude of the output is indicated by the signal V_{amp}. Figure 56(b) shows measured output

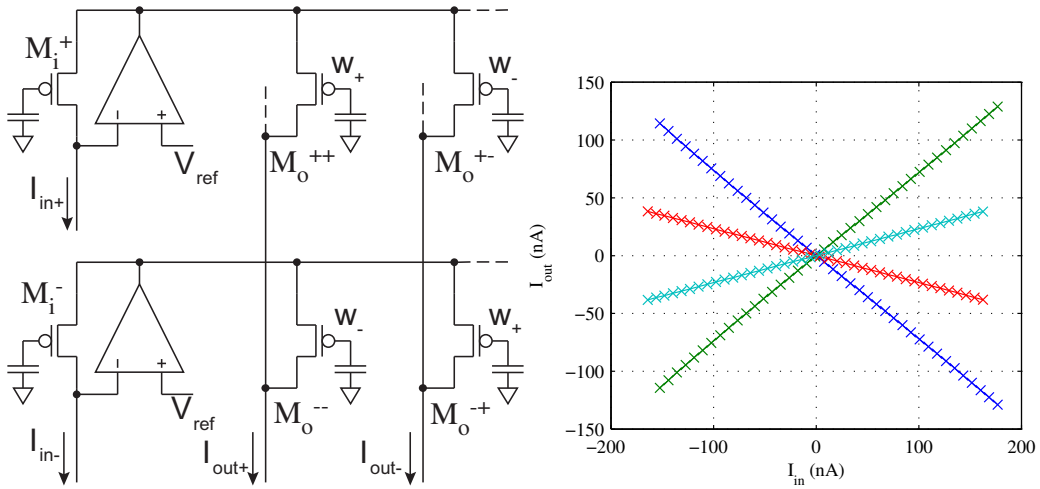


Figure 57: **Vector Matrix Multiplier:** (a) Differential circuit for 1x1 VMM to multiply an input current with a fixed weight is shown. (b) Measured data from the circuit for two different weight values.

waveforms when the input is an amplitude modulated sinusoid. The output amplitude at steady state is relatively constant. The time constants for recovery are around 11 msec. The whole circuit consumes approximately 3.76 μW of static power excluding the power consumed in buffering the signal off-chip. Figure 56(c) shows the measured gain compression characteristics. A fixed high gain and a fixed low gain characteristic is also shown for comparison. The AGC achieves 28 dB of compression for the largest input. The total variation of output amplitude is approximately 10 mV over the whole range.

6.1.7 Vector Matrix Multiplier

Unlike a Gilbert multiplier that computes the product of two time-varying signals, many applications require computing the dot product (or projection) of a time varying input vector with a fixed coefficient matrix. Prime examples of this computation are transforms (DCT, DFT) in signal processing. A fully differential circuit [24] to achieve this is shown in Fig. 57(a). The OTAs are biased at 2 μA of current. The FG devices in the circuit are routing elements, a classic example of a powerful computation being performed on the switch matrix. This circuit implements a 1X1 VMM, i.e. it

multiplies a differential input current signal with a differential weight to produce a differential output current. The core of the circuit is a floating-gate based current mirror (e.g. M_i^+ and M_o^{++}). The relation between input and output currents is given by:

$$I_{out} = I_{in} e^{\frac{\Delta V_{fg}}{U_T}} \tag{39}$$

where ΔV_{fg} is the difference in the floating-gate charge between the two transistors. To obtain four-quadrant multiplication, four such single-ended multiplications are done. The resulting differential output is given by:

$$\begin{aligned} I_{out+} - I_{out-} &= (w_+ - w_-)(I_{in+} - I_{in-}) \\ w_\pm &= w_0 \pm \Delta w, \\ I_{in\pm} &= I_{in0} \pm \Delta I_{in} \end{aligned} \tag{40}$$

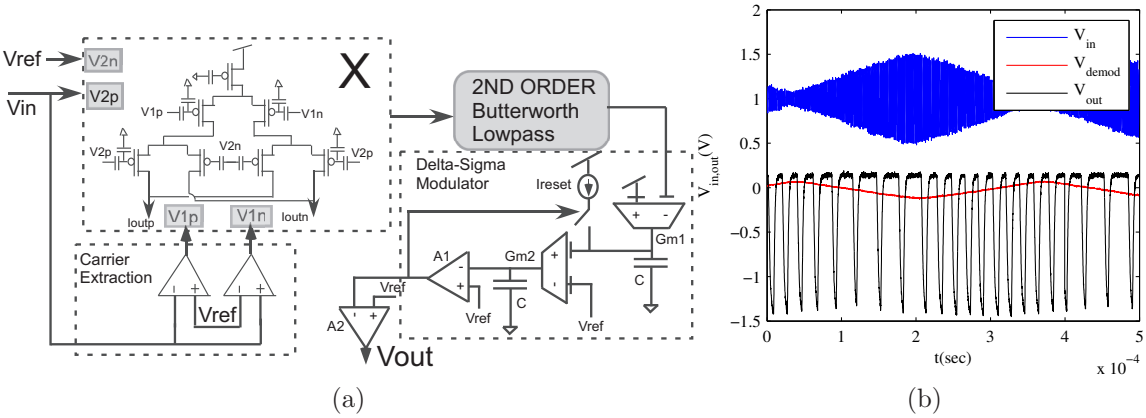
where I_{in0} and w_0 are nominal or bias values. Since the output is a current, summing many such outputs to compute the projection of an input vector on a weight vector can be achieved by KCL.

Figure 57(b) plots measured output currents from this circuit for four different weight values, two of which are of opposite signs but same magnitude. The average deviation from linearity is 7 % with the major source of error being the overlap capacitors coupling onto the floating-gate.

6.2 Larger Systems

6.2.1 AM Receiver

A receiver for an amplitude modulated signal has been implemented on the FPAA. Figure 58(a) depicts the circuit diagram for the receiver. A synchronous demodulation scheme is used to extract the modulating signal. The Gilbert multiplier is used to down-convert the modulated signal. The carrier signal is extracted from the received signal itself by using two comparators to threshold the received signal. The



(a) (b)

Figure 58: **AM baseband:** (a) Circuit for demodulation of an amplitude modulated signal and subsequent filtering and conversion to a digital bit stream. (b) Measured transient data for the whole system when the input is a 3 KHz triangle waveform modulating a 500 kHz carrier.

comparators are connected in reverse polarity to generate a pseudo differential signal. The resulting square wave signals are used to demodulate the received signal. Since a square wave at the carrier frequency is used to demodulate, a low pass filter is needed to sufficiently suppress the higher harmonic signals in the demodulated message. This is not a problem provided the modulating and modulated signals are sufficiently separated in frequency. The output of the low-pass filter is digitized using a second-order continuous-time delta-sigma modulator. The integrations of the input signal are performed using two Gm-C integrators composed of FGOTA devices. Finally the output of the second integrator is digitized by a comparator that acts like a 1 bit quantizer. This signal is used to add a reset current to the first integration node. The output digital signal is buffered using another comparator. Thus there are no clocks in this system, and the delta-sigma modulator acts as a continuous time system. The bit stream can be clocked and stored externally if desired. The dominant sources of error are the finite output impedances of the transconductors and the current sources.

Figure 58(c) shows the output of the full system when the input is a 500 kHz carrier modulated by a 3 kHz triangle wave. The demodulated and filtered waveform

shows the triangle waveform while the output of the delta-sigma modulator is a pulse-width modulated signal based on input amplitude. The static power dissipation of this circuit is around 28 μW excluding analog and digital buffers.

6.2.2 Analog Speech Processor

The components in this FPAA are suited for a variety of speech processing algorithms. Here we show an example of an algorithm for enhancing the SNR of a noisy speech signal. Figure 59(a) plots the schematic of the system along with details about one sub-band of the system. The algorithm is inspired by the physiological basis of hearing and is detailed in [21, 86]. Here we only mention the salient points of the algorithm for completeness.

An acoustic signal can be represented as

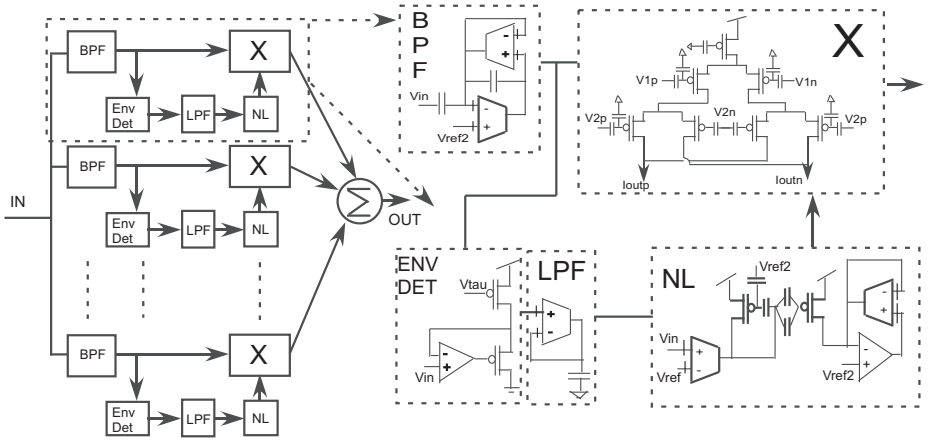
$$s(t) = \sum_n e_n(t) v_n(t) \tag{41}$$

where $e_n(t)$ is the slowly varying envelope of speech and $v_n(t)$ is the rapidly varying speech excitation component in the nth channel. The de-noising algorithm suggested by [86] requires non-linear processing of the envelope in each channel. The output of the non-linear block is related to the input as follows:

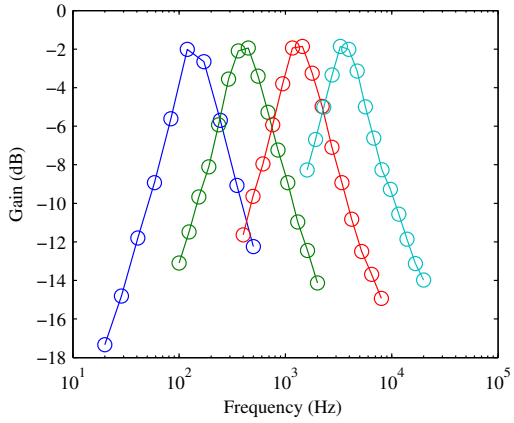
$$\hat{e}_n(t) = \beta e_n^{\alpha}(t) \tag{42}$$

Table 4: Table of Parameters

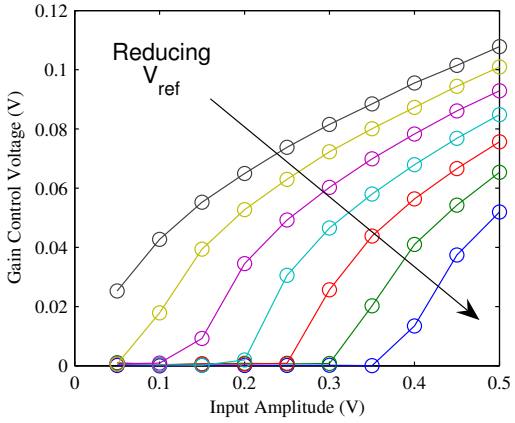
Process	$0.35\mu m$
Die Size	$3mm \times 3mm$
Power Supply	2.4V
Injection V_{dd}	$5.6V$
Number of CABs	32
Switch programming time	$N_{rows}\times$ 1 ms
Bias programming time	50 ms/element
Programming accuracy and range	9.5 bits over 6 pA to 20 μA



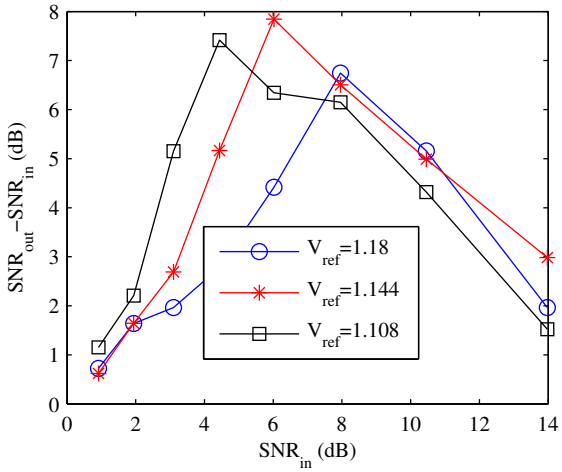
(a)



(b)



(c)



(d)

Figure 59: **Speech processor:** (a) Circuit for processing one sub-band of a noisy speech signal. (b) Programmability of the bandpass filter. (c) Gain controlling voltage input to the Gilbert multiplier for a range of V_{ref} values. (d) Output SNR for the circuits is around 7.5 dB better than input SNR.

Methods for estimating α and β are given in [86]. Intuitively, the non-linear gain applied to the original sub-band signal is like a $1 + \tanh(x - x_0)$ function with x_0 depending on the input noise level. For larger noise, the algorithm chooses a larger x_0 such that the noise is maximally suppressed while the signal still has a gain. In our implementation, the combination of an OTA and the rectifying action of a MITE based current mirror is used to create one half of a tanh function while V_{ref} sets the parameter x_0. The saturating characteristic of the other half of the tanh is approximated by a current mode square root circuit. We are currently modifying this circuit to allow for automatic generation of V_{ref}.

We present measured data for one sub-band of the proposed system. Figure 59(b) shows measured frequency responses from the tunable bandpass filter based on [43]. This shows that the results from this sub-band can be easily obtained for other sub-bands too. The total current consumption for the four cases are 1.5, 6, 24 and 96 nA. Figure 59(c) shows the input differential voltage on the gain controlling input of the Gilbert multiplier for different V_{ref} values. The non-linear processing results in almost zero gain for small signals while larger signals get larger gain. Figure 59(d) shows the output SNR improvement of around 7.5 dB over the input SNR. Also, for different input SNR values, the optimum point is reached for different V_{ref} values as expected. The static power consumption of this circuit is around 34.5 μW with 28.8 μW being dissipated in the peak detector and the TIA. Optimizations for reducing the bias current of these elements are being done currently. The equivalent digital computation for one sub-band in this case is approximately 500 kMAC leading to a computational efficiency of 14.5 kMAC/μW which is orders of magnitude better compared to an efficiency of around 2 MMAC/mW for digital signal processors [75].

Table 5: Performance Comparison of FPAA Designs

Ref.	Process	Area	Num. of CABs	CAB elements	Num. of params.	Features
This work	0.35 μm	9 mm^2	32	Prog. offset and bias OTA and multiplier, prog. bias MITE, MOSFET, Capacitor, T-gate	$\approx$ 50k[1]	Floating-gate
[30]	0.25 μm	1 cm^2	16	Integrator, multiplier, logarithm, exponent	416	log-domain current-mode
[16]	0.13 μm	1 mm^2	7	7 digitally tuned transconductors	55	Hexagonal layout
[76]	2 μm	>18.75 mm^2	10[2]	Digital and analog tuned Gm and capacitors	80	Automatic tuning
[3]	-	-	4	Switched capacitor filter, differential amp., SAR, Reference generator	16	Switched capacitor
[68]	2.4 μm	20 mm^2	4	Programmable gain, integrator/ comparator	16	Buffers for switches
[85]	1.2 μm	21.62 mm^2	5	Programmable bias integrator, amplifier, attenuator	5	Current mode

[1] All switches can be programmed in an analog way leading to around 50000 parameters. The CABs have 460 parameters.

[2] Minimum complexity of CAB considered to be 4 programmable transconductors similar to this work.

6.3 Conclusion

FPAA devices have come a long way since the introduction of the first few prototypes [65]. The RASP 2.8 generation of FPAA devices provide a powerful platform for prototyping and implementing large-scale signal processing applications. Table 4 presents the parameters for this chip. The programmable switch matrix composed of floating-gate devices shows excellent isolation and can be readily utilized in computation. Different levels of routing allow implementation of high performance circuits while allowing for fast turn-around times. A comparison with other FPAA chips is presented in table 5. To the authors' knowledge, there is still no concrete performance metrics to compare different FPAA designs. One method that may be used in the future is to have a number of benchmark applications (filters, vector-matrix multiplication etc.) for which power per unit computation or speed may be compared. In that case a possible figure of merit is:

$$\begin{aligned} FOM &= \frac{(\text{Number of parameters})(MMAC/\mu W)(f_{max}/f_T)(SNR)}{(Area/L_{min}^2)} \\ &= \frac{(\text{Number of parameters})}{Area}\frac{f_{max}L_{min}^2}{f_T}(MMAC/\mu W)(SNR) \end{aligned} \tag{43}$$

where f_{max}, SNR and $MMAC/\mu W$ are the maximum bandwidth, signal to noise ratio and power efficiency obtained for the benchmark applications while f_T and L_{min} are features of the VLSI process used. The chip we present is the largest design reported with around fifty thousand programmable analog parameters and has significantly more variety in CAB components compared to others. We hope to use this chip for prototyping and implementing systems for a variety of applications ranging from speech processing to sensor interfacing.

One of the most important improvements of this chip over its predecessor FPAA designs is the on-chip integration and improved performance of the FG programming circuitry. The on-chip programming interface allows current measurements from 6 pA to 20 μA . The fully digital interface allows easy integration with a microprocessor.

Programming times are around 50 ms for accurate biases and 1 msec per row of switches. In the next couple of chapters I shall describe the circuits that enable this.

CHAPTER VII

THE MAGIC ELEMENT: FLOATING-GATES AND PROGRAMMING THEM

Floating-gate transistors have been used in many large scale VLSI systems as multi-level digital memories, neural-network synapses or reconfigurable switches in field programmable arrays. We present a generic architecture for programming floating-gates over a wide range of currents at moderate accuracy and speeds. Moreover, the fully digital interface allows easy integration of the floating-gate chips in a larger embedded system. This system allowed programming floating-gates to currents in the sub pA range which is essential for achieving the long time constants needed for modeling biology.

The basic architecture for programming is similar to the one described in [93] with selection circuitry on the periphery of the array and measurement circuits for the currents placed along rows or columns. The speedup compared to [93] is achieved primarily by reducing the time needed to measure currents which is generally the bottleneck. The large measurement range is achieved by using a logarithmic transimpedance amplifier as compared to a fixed transimpedance gain as in earlier versions [8]. The output voltage of this topology provides a temperature independent measurement of the floating-gate potential. Also, digital-analog and analog-digital converters have been included on the chip making the entire interface digital and easily controllable by a microprocessor. This chapter provides complete measurement results of the programming system from 0.5 μm and 0.35 μm CMOS and describes final accuracy and dynamic range achieved in programming floating-gates by this method.

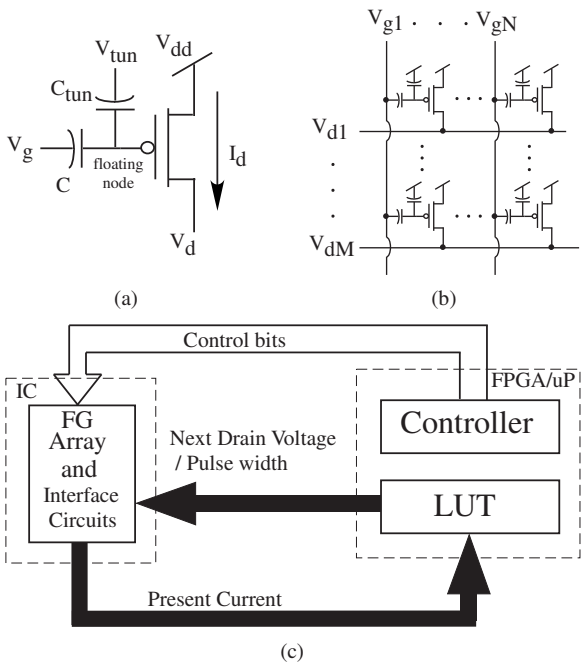


Figure 60: **Floating gate programming system:** (a)A single Floating gate(FGMOS). (b) An array of floating gates. Only selected FGMOS has a low value of gate and drain voltages enabling injection while others have either of the voltages turned to V_{dd}. (c) Data flow diagram for the whole system comprising the IC and the digital control on the FPGA or microprocessor.

7.1 Overview of Programming Floating gates: First Principles

Floating-gates can be programmed by both Fowler-Nordheim tunneling and hot-electron injection processes. Using tunneling as the method of charge movement significantly increases programming time with higher desired precision because of the logarithmic behavior of the mechanism [58]. Hence tunneling is used as a global erase while hot-electron injection is employed for fast, accurate programming of these elements. Figure 60(a) depicts a floating gate with its terminals marked. The process of accurately programming an FG-transistor consists of two distinct phases: 'measure' when the current through the devices in the array are measured and 'inject' when the devices are injected to reach their desired targets. To inject a device, all the terminal voltages, i.e. V_{dd}, V_d, V_g and V_{tun} are raised to a value higher than the normal operational value but their relative values are kept same. This process is referred to

as ramp-up. In the current generation of chips, this high voltage is generated using an on-board DC-DC converter but will be replaced by an on-chip charge pump in the future. The high electric field necessary for injection is produced by pulsing the drain to a lower voltage for a certain time(t_{pulse}).

Figure 60(b) shows an array of these elements. To select a device, an enabling voltage is applied to its gate and drain terminals. All other elements have either of the gate or drain voltages set to V_{dd} thus prohibiting injection. This condition is a direct application of the fact that ample source current and large drain-channel potential are both necessary for hot-electrons to inject onto the gate. In the chip fabricated in 0.35 μm, the source current is cut off in the non-selected devices by an explicit switch. Figure 60(c) shows the data flow in the automated programming system. The digital word corresponding to the present current (obtained during a 'measure cycle') of the selected floating gate is used to index into a Look-up table(LUT) on the FPGA, PC or microprocessor(μP). This LUT has values of next drain voltage or t_{pulse} based on the algorithm used to program the floating gate. This value is used for the next programming cycle ('inject cycle') and the process is iterated till desired accuracy or some other stopping condition is reached. Instead of a LUT, coefficients of a polynomial fit to the injection characteristics may also be used [8]. The controller for sequencing these operations and selecting desired gates is also implemented on the FPGA/PC/μP.

Before concluding this section, we provide a brief description of one possible programming algorithm, details of which are available in [8]. This algorithm modulates the drain-source potential while maintaining a fixed pulse width. It is shown in [8] that the change in FG current due to injection can be modelled as:

$$log(\Delta I) = m(I_{init})V_{ds} + f(I_{init}) \tag{44}$$

where I_{init} is the current in the FG device prior to injection, ΔI is the change in the FG current after injection and m, f are polynomials (typically quadratic). Initially,

a random set of FG devices in the chip are subjected to hot-electron injection for different V_{ds} values to obtain a mean characteristic. This function can now be inverted to obtain a desired V_{ds} for a certain value of I_{target} and I_{init}. This continuous function can also be discretized using a desired grid-size to generate a LUT indexed by I_{init} and ΔI. Thus ideally, we would need to access the LUT only once. However, because of mismatch, the injection parameters vary across FG elements. To ensure that the FG current does not exceed I_{target}, at every step either V_{ds} or t_{pulse} is reduced from the computed one by a factor determined by mismatch. This leads to a tradeoff between programming time and accuracy/mismatch. A similar algorithm may be formulated for pulse width modulation.

7.2 On-chip Programming: Architecture and Timing

In this section we discuss the architecture of the on-chip programming system. The system has been tested as a separate chip and also as part of a larger FPAA IC. The architecture is general and can be employed to program floating-gates in other systems as well.

Figure 61(a) shows the generic architecture of a system to program an array of floating-gates. The floating-gate(s) to be currently programmed are selected by applying a digital word to the selection circuitry on the periphery of the array. The selection may be done one at a time or a row at a time or in any other parallel fashion as desired. As the programming process involves ramping up the terminal voltages of the floating-gates, the control signals must also be referenced to the present V_{dd} of the chip. Hence, level shifters are included which convert the 3.3 V digital signals to the level of the programming V_{dd}. The selection circuit passes the desired gate and drain voltages to the selected FG device, while the gate and drain terminals of all the rest are set to V_{dd}. The source of all floating-gate transistors are tied to V_{dd} and do not have any selection mechanism. The tunneling voltage connection that goes to

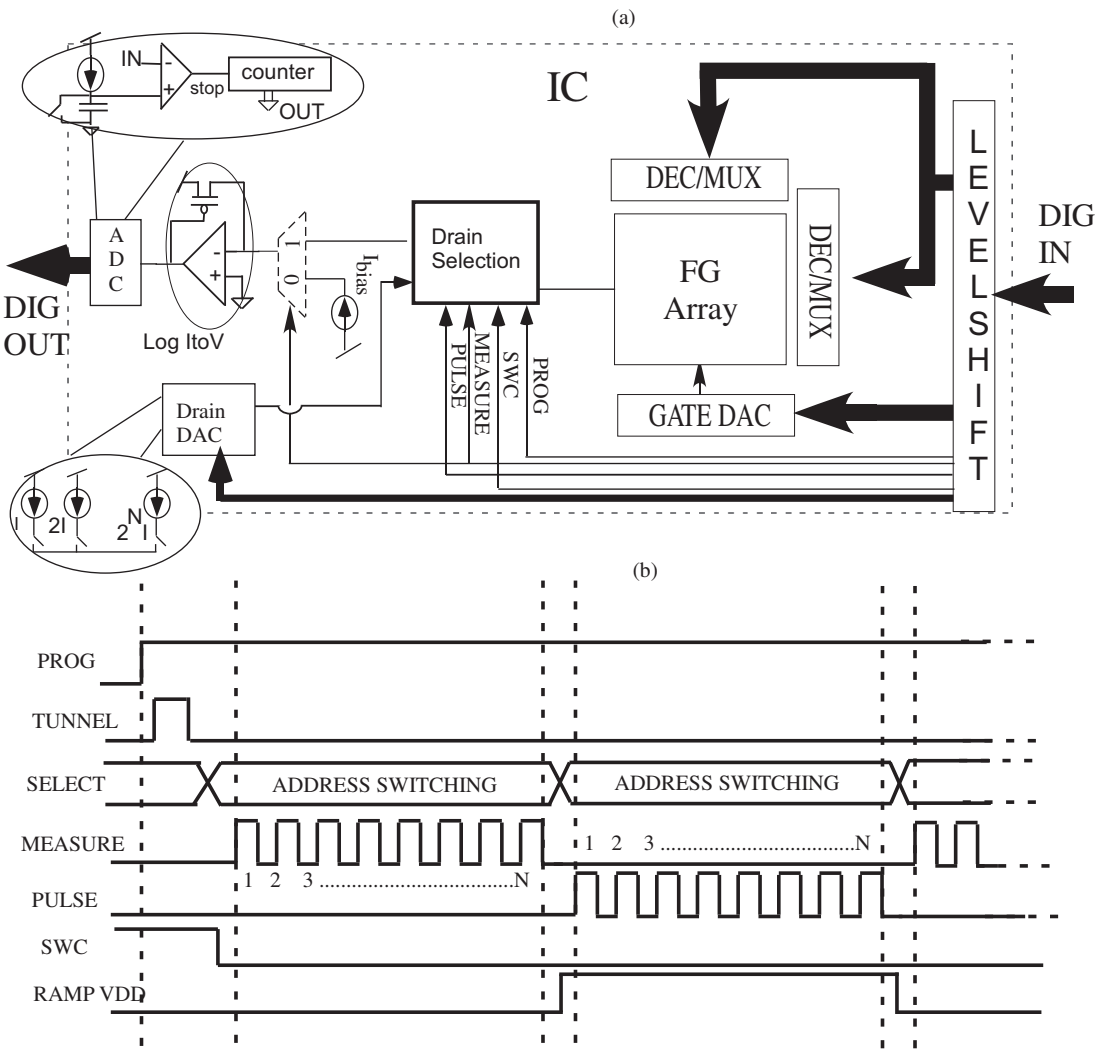


Figure 61: **System Architecture and Timing:** (a)Architecture of the programming system showing decoding and multiplexing logic to select desired floating gates from an array as well as the data converters for interfacing with the FPGA/μP. (b) Timing diagram of the chip showing sequence of signaling for programming 'N' floating-gates accurately (the time durations are not to scale).

all FG devices is not shown in this figure. The gate and the drain DACs supply the desired voltages to the gate and drain of the FG elements and are controlled digitally through an SPI interface. While the drain DAC is used only during injection, the gate DAC is used in both programming and operational mode.

The measurement of the charge on the gate is accomplished by measuring the current through the device using a logarithmic transimpedance amplifier(TIA). The logarithmic compression allows the TIA to measure currents varying over several

decades in magnitude. The amplifier maintains stability without dissipating excessive power by employing an adaptive biasing scheme that will be described later. The output of the amplifier is low-pass filtered and then digitized using a ramp ADC. The ramp topology is chosen because of its linearity, ease of implementation and because the long conversion time of the ADC is not the bottleneck in determining speed. The bottleneck in this case is the settling time of the TIA for low-current inputs. The combination of the logarithmic TIA and the linear ADC form a floating-point ADC as will be explained later.

The four major control signals determining the operation of the architecture are described now:

1) PROG: This signal being high indicates that the FG elements in the chip are being programmed to the desired value. When it is low, the chip is in operational or RUN mode.

2) MEASURE: This signal defines a sub-mode for PROG mode. MEASURE being high indicates that the system is in 'measure' mode, i.e. currents of the programmed gates are being measured using the floating point ADC. On the other hand, MEASURE being low signifies that the system is in 'inject' mode, i.e. gates are being injected to reach the desired target values.

3) PULSE: This signal is high when the floating gates are being injected. The pulse-width of this signal determines the time, t_{pulse} for which the selected gate is injected in the current cycle.

4) SWC: This signal is asserted high when the chosen floating gate needs to be programmed as an ON switch, i.e. it needs to be programmed to an arbitrary low floating gate voltage.

Fig 61 (b) shows the system timing diagram for programming 'N' FG element accurately. After PROG is asserted high (signaling the beginning of programming mode), a tunneling pulse is used to globally erase all the array elements. SWC is then

asserted low indicating accurate injection mode followed by selection of the desired element or row of elements. The 'measure' phase begins first where the charge on all the floating gates are measured. The MEASURE signal needs to toggle for every element as it marks the beginning of the ADC conversion cycle. After the 'measure' phase, the chip is ramped up followed by 'N' short pulses on the signal PULSE. If pulse-width modulation is used to program the elements, then the duration of each of these pulses are as computed by the algorithm for reaching targets. If drain voltage modulation is being used, the drain DAC sets the drains to desired voltages before every pulse and width of the pulses on the PULSE line are fixed. This is followed by ramping down the chip followed by another 'measure' cycle and so on.

For programming the switches, the control sequence is simpler as measurements are not needed. In that case, SWC is asserted high to indicate switch programming mode and MEASURE is kept low throughout the process. The rest of the signaling is as described earlier.

7.3 On-chip Programming: Components

In the last section, the architecture and global signaling scheme was detailed. In this section, the major components of the system, i.e. the drain selection block, the drain and gate DACs, the logarithmic TIA and the ADC are discussed along with measured results from 0.5 μm and 0.35 μm chips.

7.3.1 Drain Selection

The drain selection circuitry as shown in Fig. 62 acts as a second selection level after the desired floating-gate drain terminal has been selected by multiplexors. This block switches the selected drain to injection or measurement sub-circuits depending on the programming mode. If the system is in measure mode, the drain is connected to the transimpedance amplifier. In inject mode, if PULSE is low, the selected drain is tied to V_{dd} thus prohibiting injection. When PULSE is high, the selected drain is

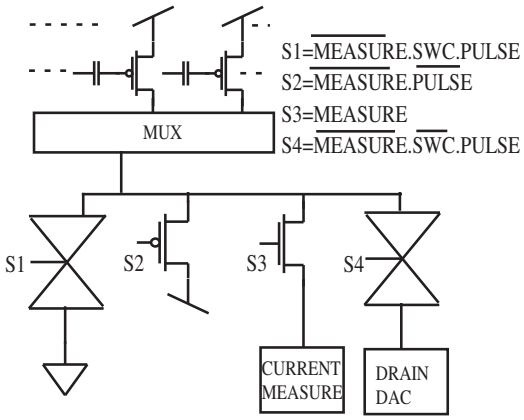


Figure 62: **Drain Selection:** The drain selection circuit connects the drain of the selected FG device to the DAC, V_{dd} or ground in inject mode and to the measurement circuitry in measure mode.

switched to the drain DAC or to ground depending on the polarity of the signal SWC. This is because for switch programming, it is always desirable to have the maximum Vds across a selected device, while in accurate programming, the Vds is modulated depending on difference from target current.

7.3.2 Gate and Drain DAC

The gate and the drain DACs share a binary current scaled architecture as shown in Fig. 63(a). The reason for this choice was the low required resolution of 7 bits for either DAC. We do not need very high resolution for the drain DAC since we can trade-off the time needed for injection with the number of possible drain voltage levels. The gate DAC's resolution can also be low, since, it is used to set the operating regime of the FG transistor being injected within a *range* of subthreshold currents that correspond to high injection efficiency. The current sources are cascoded PFETs biased by a proportional to absolute temperature(PTAT) bootstrap current source while the cascode transistors are biased by the structure described in [72]. The sizing of the current source array was done following [110] and the references therein. To guarantee operation, the devices were chosen large enough to satisfy 8-bit matching. Dummy devices were employed to eliminate systematic mismatch. The resulting area of the DACs is around 50 % of the entire area of the programming infrastructure. A

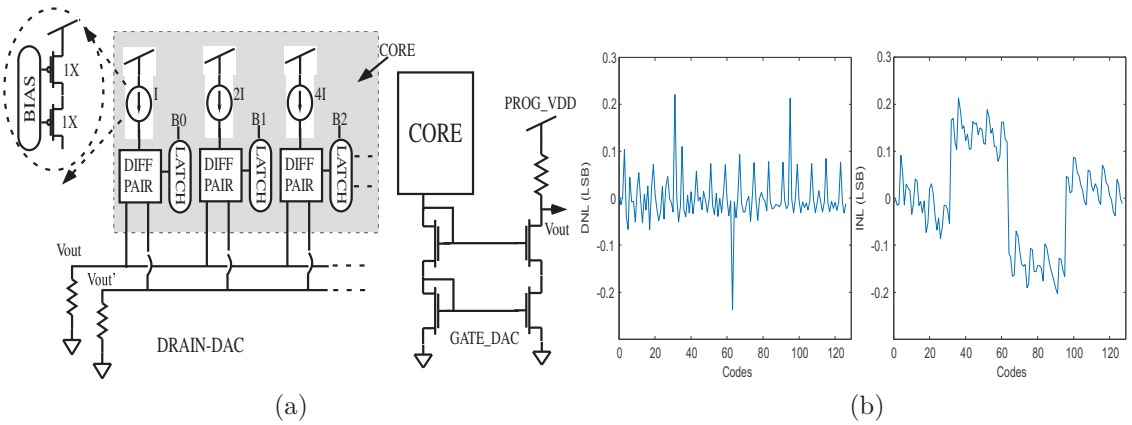


Figure 63: **GATE and DRAIN DAC circuit and measurements:** (a) Both the DACs have a binary current scaled architecture and have a resolution of 7 bits. The output currents are converted to voltage using resistors giving a maximum swing of 1V in the drain DAC and 2V in the gate DAC. The gate DAC needs to provide voltages close to the programming V_{dd} and hence the current is mirrored to create a voltage referenced to V_{dd}. (b) Measured DNL and INL of the DACs from a 0.35 μm chip. Both the DACs exhibited better than 7 bits of matching as expected.

differential pair is used to switch the currents to increase switching speed. A latch is used to convert the 3.3 V digital signals to smaller voltage swings with a low crossing point [110] so that the ON transistor of the differential pair is in saturation.

The drain DAC needs to provide voltages close to ground and hence the currents are directly passed into a resistor. The voltage drop across the resistor does cause early voltage induced errors, but since monotonicity was the most important criterion in this application, it was not found feasible to include an amplifier to fix the output node. The gate DAC, on the other hand, needs to provide voltages close to the programming V_{dd} while it is powered from a separate 3.3V supply that is common to the programming circuit. So the current was mirrored using a cascoded NFET mirror and passed into a resistor referenced to the programming V_{dd}. The digital words for the DACs are loaded into a shift register from the digital controller through a SPI interface.

Figure 63(b) shows the measured DNL and INL from the DAC structures fabricated in a 0.35 μm chip. The matching of the transistors was better than 7 bits as expected

7.3.3 Adaptive Logarithmic Transimpedance Amplifier

The logarithmic amplifier is the most critical element in this programming architecture. In fact, it provides a solution to the general problem of wide dynamic range current measurement with low power dissipation and will be detailed separately in the next chapter. Here, I shall just briefly mention a few salient points of the circuit.

Figure 64(a) shows the transimpedance structure that has a variable resistance in the feedback path across an amplifier. The amplifier tries to hold the input node constant, forcing the current to flow through the feedback transistors M1 and M2 and reducing the current through the capacitance at the input node(C_{IN}).

To measure a wide dynamic range of currents spanning several orders of magnitude, a fixed feedback resistance should have a very small value, which poses an SNR issue at low currents. On the other hand, a MOS transistor changes its resistance depending on the current flowing through it. To increase the sensitivity of the conversion, M2 is used as a source degeneration for M1.

To maintain stability, the amplifier's output pole should be at a frequency that is sufficiently higher than the pole at the input due to C_{IN} and any parasitic/explicit feedback capacitor C_F. To achieve this, most approaches burn excessive power because the amplifier is biased with currents that are much higher than the highest input current to guarantee stability in the entire operating range of currents. In our implementation, M3 and M4 replicate the input current, while M5 mirrors the current into the amplifier's bias with a gain. Thus, we burn less power in the amplifier when the input current is low. The same reference voltage is used for the amplifier and the adaptation circuit.

An advantage of this topology is that when the TIA is measuring currents from a floating gate, its output voltage is effectively measuring the floating-gate potential (because the current is exponentially related to floating-gate potential). This leads

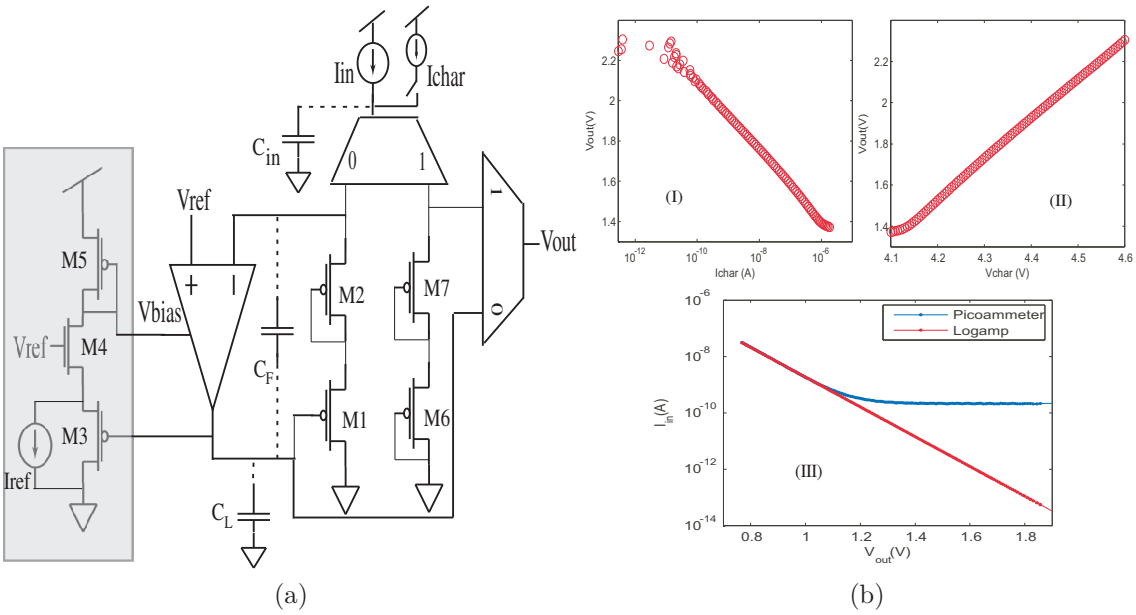


Figure 64: **Logarithmic TIA circuit and measurements:** (a) The logarithmic TIA employs PMOS transistors, M1 and M2 in feedback across an amplifier. The exponential I-V relation of a MOS in subthreshold allows the logarithmic relation. The adaptation block in the shaded area modifies the amplifier's bias current according to input and helps maintain its stability over a wide range of inputs. M6 and M7 form an open-loop I-V converter that is used for very high currents which make the other structure unstable. (b-I) Measured data from a 0.5 μm chip showing the input current and the output voltage of the TIA. The current was created by sweeping the gate voltage of the PMOS implementing the current source I_{char} in Fig. 64. The deviation from logarithmic behavior at high currents is due to transistors entering above threshold regime and at low currents due to the pico-ammeter losing accuracy. (b-II) The same data as (b-I) plotted against the gate voltage of the PFET, V_{char} that produces I_{char}. Here the plot is linear even at very low currents (high gate voltages) confirming the fact that the on-chip logamp is more accurate than the off-chip pico-ammeter for very low currents. (b-III) Measured data from 0.35 μm chip again showing improved accuracy of the logarithmic amplifier over off-chip instrumentation.

to the following expression for the output voltage of the TIA:

$$V_{\text{out}} = \frac{V_{\text{ref}}}{\kappa^2} - \frac{V_{dd} - \kappa V_{fg}}{\kappa_{eff}} - \frac{U_T}{\kappa_{eff}} ln(M) \tag{45}$$

where V_{fg} is the floating-gate potential and the FGMOS has an aspect ratio that is M-times the aspect ratio of the feedback transistors. Thus by appropriately sizing the feedback transistor, the output of the TIA can be made temperature insensitive to a first order since the charge on the floating-gate does not change appreciably with temperature [92].

The difference in the structure here as compared to the stand-alone one described

in [14] is the added multiplexors and diode connected PFETs M6 and M7. These were added based on system considerations since the feedback circuit described earlier loses stability at very high currents. So, M6 and M7 is kept as a coarse I-V converter for high currents. The output of the two converters are multiplexed based on the signal SWC as only transistors programmed as switches might produce such high currents. The current source I_{char}, implemented by a PMOS, is kept to bias the circuit when it is not measuring currents. The source or gate of the PFET is controlled by a DAC on the board and is also used for characterizing the performance of the TIA.

Figure 64(b) shows measured characterization data from both 0.5μm and 0.35 μm IC designs. Figure 64(b-I) shows the I-V relation for a logarithmic amplifier fabricated in a 0.5μm process. The input current was created by sweeping the gate voltage of the PMOS used to create I_{char} in Fig. 64(a). The deviation from logarithmic relation at high currents is due to transistors entering above threshold region of operation. At low currents, the off-chip pico-ammeter (Keithley 6485) loses accuracy. Figure 64 (b-II) shows the same data but plotted against the gate voltage of the PFET used to create I_{char} on the X-axis. The logarithmic relation is now maintained for very low currents too (high gate voltage). This conclusively shows that the logarithmic TIA is more accurate than the off-chip pico-ammeter for lower currents. Figure 64 (b-III) shows similar data from a 0.35 μm FPAA chip. Here, the measured data between currents of 1 to 10 nA is used to fit a polynomial to the logamp characteristic. Using this curve-fit, the output voltage of the logamp is used to infer measured currents and compared with the pico-ammeter measurements. While the off-chip measurements saturate at around 100 pA due to noise and leakage from ESD diodes, the on-chip measurement can go down to sub-pA levels proving its utility. Conformance to logarithmic behavior can be measured by fitting a line to the characteristic. The average error is 3.1 % (considering currents less than 1 μA) because of the feedback transistors entering the above threshold regime at high currents. However, using a second-order fit, the error

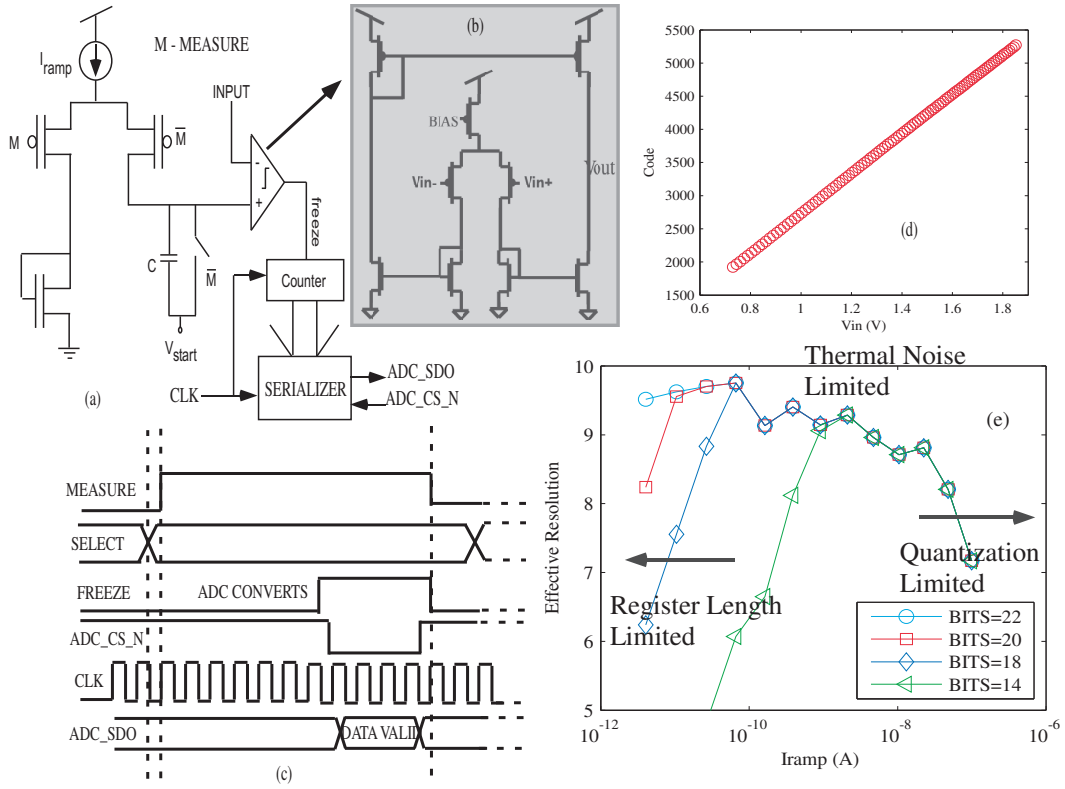


Figure 65: **ADC Schematic:** (a) Simplified schematic of the ramp ADC. The signal M represents MEASURE. The output of the counter is passed to the controller through a SPI interface. (b) The comparator is a nine transistor OTA. For higher gain, cascode transistors are being used in the output stage in future versions. (c) The timing diagram of the ADC showing that after the desired floating gate is selected, MEASURE starts the conversion process. Once FREEZE, the output of the comparator goes high, the output of the counter is serialized and sent to the digital side through an SPI interface. (d) Measured input-output relation of the ADC in 0.35 μm CMOS. The input of the ADC was set by passing a certain current through the TIA. (e) Measured resolution of the TIA-ADC combination is plotted against varying ramp current, I_{ramp}. For very fast ramp, the quantization noise dominates the performance while for very slow rates the finite counter bit-length dominates. In between the two regimes is the thermal noise limited part.

reduces below 1 % for the same range. Using a high order fit for the I-V conversion is not a problem since this can be done on a PC before sending the target voltages/codes to the on-board μP.

7.3.4 Ramp ADC

The ADC in this system acts as the interface between the TIA and the digital controller. As the settling time of the TIA for sub-pA currents is of the order of a few msec, the conversion time requirement for the ADC is also relaxed. This led to the

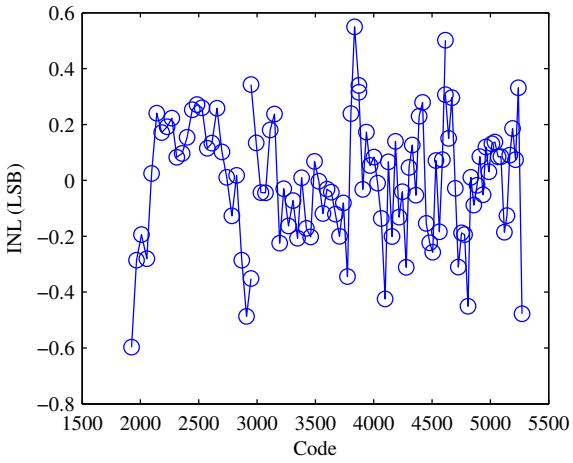


Figure 66: **ADC linearity:** The measured INL of the ramp converter is plotted with respect to the LSB of a 9-bit ADC. To improve the linearity, the range of codes is divided into four ranges and separate gain and offset correction factors are stored for each range.

choice of a ramp ADC architecture as shown in Fig. 65(a) because of its simple structure. In the figure, M denotes the control signal MEASURE starting the conversion and V_{start} is the starting voltage for the ramp. The comparator trips after the ramp generated by the current source, I_{ramp}, crosses the input voltage freezing the counter. When the conversion starts, there is a shift in the start voltage of the ramp due to charge injection from the switch. But this is signal independent and hence can be treated as an offset. It can be taken care of by either offsetting V_{start} (analog trim) or by subtracting the digital word corresponding to the offset (digital trim). The accuracy of the ramp is limited by the early effect of the cascode current source used. The biasing of the cascode is done following [72] while the current source is biased using a bootstrap current source chosen because of ease of implementation and relatively low temperature dependance (PTAT in subthreshold operation). It can be replaced by a lower TC current source in the future. The comparator in Fig. 65(b) is a simple high-gain amplifier comprising a differential pair followed by a push-pull output stage. The gain of the comparator will be increased in future versions by employing cascode transistors in the output stage.

Figure 65(c) shows the details of the timing of the digital interface for the ADC.

Once the output of the comparator, FREEZE goes high, the counter has the valid digital word at its output. The chip-select signal, ADC_CS_N is then asserted low and the serialized data is read out from ADC_SDO. Once the SPI data transfer is completed, MEASURE is asserted low and the ADC is ready for the next input. In this implementation, a 14-bit counter has been used. The FREEZE output is also buffered out allowing it to be used to control off-chip counters on the μP which can be 32 bits long.

The effective resolution for the TIA-ADC combination, N, can be found by relating it to the dynamic range. Let the logarithmic amplifier's characteristic be given by V=Offset+K×log(I) and δV be the voltage noise level at the output of the logamp. Then denoting the input referred noise current to be δI when the DC input current is I, δV can be related to the SNR of the input as:

$$\delta V = K\frac{dI}{I} = \frac{K}{SNR} \tag{46}$$

Then, using (46), dynamic range(DR_{ADC}) of the ADC is given by:

$$\begin{aligned} DR_{ADC} = 2^N &= \frac{V_{max} - V_{min}}{\delta V} \\ &= \frac{Klog(\frac{I_{max}}{I_{min}})}{\delta V} \\ &= \frac{I}{dI} log(\frac{I_{max}}{I_{min}}) \\ &= SNR \times log(DR), \end{aligned} \tag{47}$$

where DR is the dynamic range of input currents. This equation explicitly shows the floating point nature of the system as the SNR sets the mantissa bits while log(DR) sets the exponent. The number of exponent bits are around 3-4 for 5-7 decades of current. The clock frequency can be decided considering the worst case conversion time to be $2^N \times T_{clk}$, where N is the number of bits and T_{clk} is the clock period. The maximum frequency of the clock available from the μP was around 20 Mhz, leading to $T_{clk} = 50$ ns. For N equal to 14, the worst case conversion time is around 1 ms and

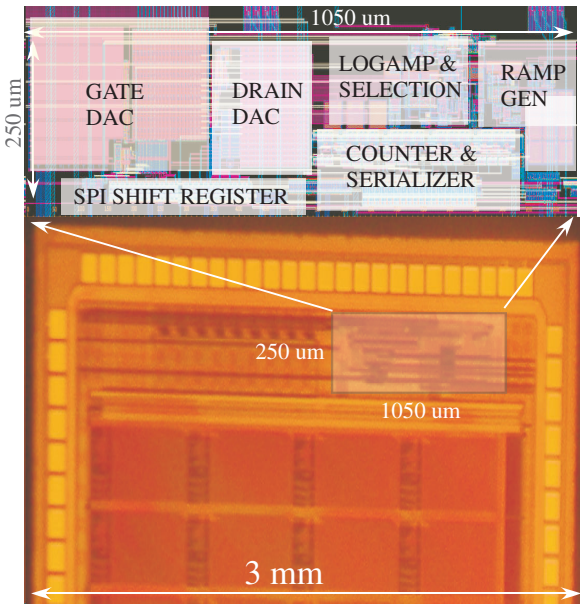


Figure 67: **Die photo:** Die photo and layout of the 0.35 μm chip showing the different sub-blocks of the programming infrastructure.

average conversion time is around 512 μs. The value of the current source can then be chosen using the following equation:

$$V_{LSB} = \frac{I \times T_{clk}}{C}, \tag{48}$$

where V_{LSB} is decided by the noise at the output of the I-to-V and C is 5 pF.

Figure 65(d) shows measured transfer characteristic of the ADC with a 32 bit counter implemented on the μP. This was done to examine the effect of finite register length on the ADC output shown in 65(e) and explained later. The input voltage of the ADC is swept by varying the current through the TIA. In this chip, the digital trim option was preferred and V_{start} was set to ground to avoid using an extra pin. Thus, a part of the counter's range was sacrificed in the process as the output of the TIA does not start from ground. This is evident from the count not starting from zero in the figure. It has been found that even though the TIA can measure currents down to sub-pA levels (found by monitoring the voltage output), the ADC cannot convert inputs lower than 6 pA reliably. This is traced back to the fact that the digital signal starting the conversion also starts the measurement phase for the TIA. For very low currents, the TIA output does not settle in time. This has been rectified in future

version by having separate digital controls and employing a bidirectional logamp for faster settling.

The effect of reducing the quantization noise by slowing the ramp has also been studied. Figure 65(e) shows the measured resolution corresponding to different ramp currents, I_{ramp}. In this experiment, the current through the TIA is set to a fixed value producing a fixed voltage at the input of the ADC along with some noise. For a particular I_{ramp}, this input is digitized multiple times and the ratio of the mean of the codes and their standard deviation is considered as SNR for computing the effective resolution. It is seen for very large I_{ramp}, the LSB voltage step of the ADC is too large leading to a quantization noise dominated performance. On the other hand, for very slow ramps, the number of bits in the counter, CNT, is the limiting factor giving poor dynamic ranges. In between these two regimes, performance is limited by the noise of the system contributed primarily by the noise in the log-amplifier's output and in the current source generating I_{ramp}. 9.5 bits of performance corresponds to a rms noise of around 1.5 mV at the output of the TIA and around 80 μV on the floating-gate. Performance can be improved up to 11 bits by averaging a number of readings. This is done in FG programming when the measured current is close to the desired current.

Figure 66 plots the INL of the ramp converter with respect to the LSB of a 9 bit ADC. The finite output resistance of the current source leads to nonlinearity in the slope of the converter. To ensure that the error in linearity of the ramp is less than 0.5 LSB, digital correction is used. The entire range of codes is divided into four sections and a separate gain and offset correction factor for each section is stored in the μP. To predict the correct input voltage for a certain code, one of these four gain and offset factors are used based on the range in which the code falls. Dividing the range into more subsections can improve INL performance even more, but is not needed since noise dominates the ADC performance in that case.

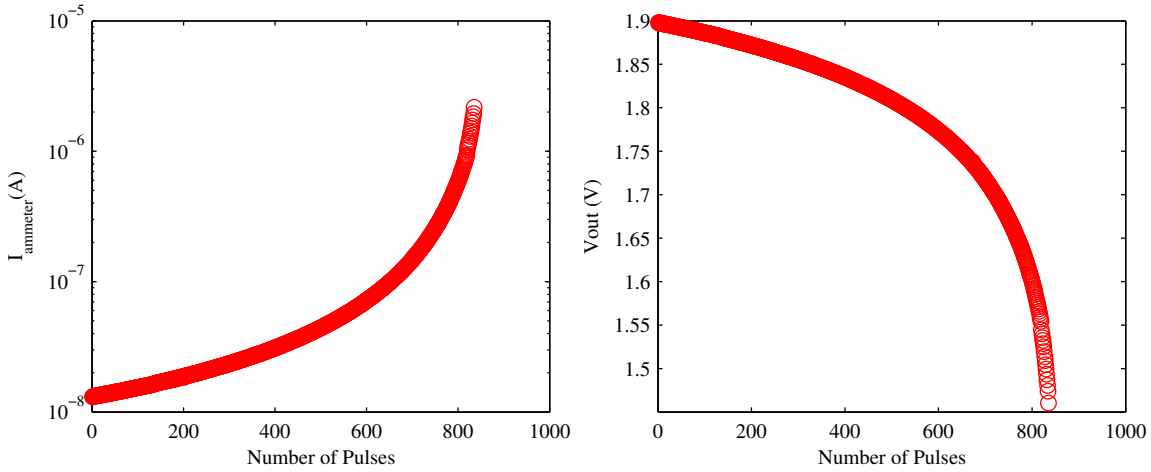


Figure 68: **Floating-gate injection:** Measurements from a 0.5μm chip change in floating-gate current with 100μsec injection pulses of Vds equal to 6.5V. (a) plots ammeter measurement and (b) plots the output of the logarithmic TIA.

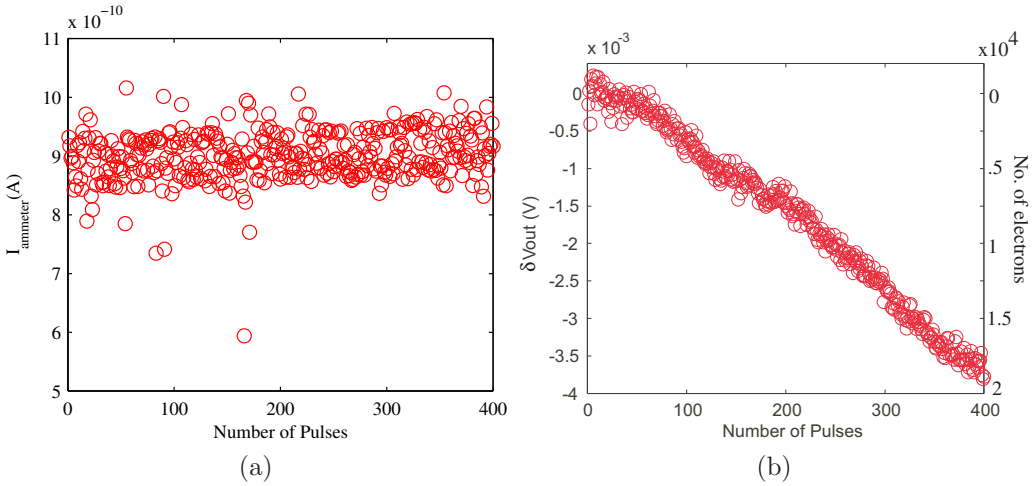


Figure 69: **Measuring injection accurately:** 100μsec wide pulses of Vds, 6V in magnitude, were applied to a floating-gate in 0.5 μm and the current was recorded using (a)picoammeter and (b)TIA. Clearly the on-chip TIA is able to detect the injection while the ammeter is not. The left axis on the plot is the change in voltage from the starting point, 2.085V. The right axes shows the equivalent number of electrons moved on the gate.

7.4 Floating-gate Measurements

In this section, we describe system test results showing measured change in floating-gate charge and using that information to program a desired amount of charge. Figure 67 shows a die photo of the 3×3 mm^2 chip fabricated in 0.35 μm CMOS with the programming infrastructure occupying 1050×250 μm^2. A close up of the layout of the

programming circuits is also shown. Figure 68 shows measured results of floating-gate injection. In this experiment, a floating-gate was subject to 100 μsec pulses of Vds equal to 6.5 V. After every pulse, the current from the FG device was measured using the TIA and an ammeter (Keithley 6485). This figure plots the ammeter reading in (a) and the output of the TIA in (b) with number of pulses showing injection increasing the current. Figure 69 shows data from another experiment where the FG device was subjected to 6V Vds pulses of pulse width equal to 100 μsec. The current was monitored after every pulse using both the on-chip TIA and the off-chip ammeter. From the measurements, it is obvious that while the TIA can distinguish very fine amount of hot-electron injection, the ammeter cannot. The left axes shows change in voltage from starting point while the right axes of the plot shows number of injected electrons based on a gate capacitor of value 750 fF.

Next we show programming floating-gates to specific target currents that span a wide range in magnitude. A set of forty floating-gates are programmed to currents ranging from approximately 6 pA to 20 μA using a version of the algorithm shown in [8]. The experiment was run fifteen times choosing a random set of devices from a pool of over thousand devices. The average of the absolute error in achieving the current targets is plotted in Fig. 70 (a). The average error is 2.14 % for this range of currents and reduces to below 1 % if currents higher than 100 pA are considered. There could be several reasons for this error; first, there is an error associated in modeling the injection process over different source currents and V_{ds} values by a polynomial. This error is magnified particularly when the dynamic range is large. Second, there is a spread of the parameter values for injection across a large number of devices. Both these errors can be handled by slowing down the rate at which the target is approached and averaging the measurements to reduce noise. The increased error at low currents is primarily due to more noise, a property associated with logarithmic amplifiers followed by a fixed-bandwidth low-pass filter [14]. This issue

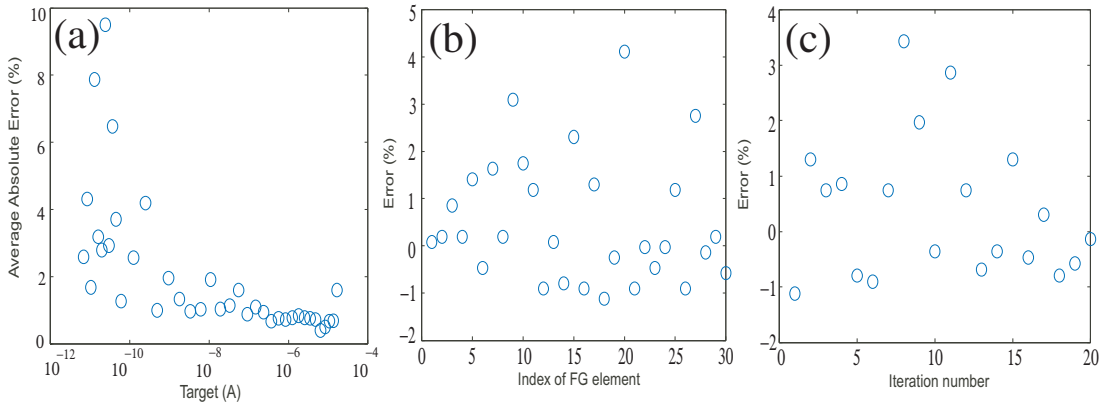


Figure 70: **Error in programming:** (a) The result of programming a set of floating-gates to currents varying from 6 pA to 20 μA is shown. The results are averaged over fifteen iterations of the same experiment but choosing a random set of devices for every iteration. The average error is less than 1 % in the range of around 100 pA to 20 μA. (b) The resulting error for a set of thirty different devices programmed to 100 nA. (c) The resulting error when a single device was erased and programmed twenty times to a target of 100 nA.

can also be addressed by averaging the measurements and trading speed for accuracy.

Figure 70(b) shows the error when a set of thirty different devices were programmed to a current of 100 nA while Fig. 70(c) depicts the resulting error when a single device is tunneled and programmed for twenty times to achieve a target of 100 nA every time. The source of this error is also the noise in the current being measured added to the noise in the measurement.

7.5 Discussion

7.5.1 Long-term Storage

The data retention capability of floating-gate transistors have been reported in multiple publications [1, 100, 101] and exhibits insignificant charge loss in tens of years. Both short-term and long-term drift of charge in our floating-gate devices have been shown to be less than 0.2 percent over sixteen days [99]. Moreover, the charge drift has been observed to reduce after an initial bake in the oven at elevated temperatures [101]. Finally, the effect of charge drift on circuit performance depends on the topology of the circuit and might be reduced if differential measurements are taken.

Table 6: Performance Comparison of FG programming

Ref.	Method	Range	Prog. Accuracy	Prog. Time	Level of integration	Array Programming
This work	Discrete pulses, CHE injection	6 pA - 20 μA ($\Delta V_{fg} \approx 0.63$ V)	9.5 bits (floating point)	50 ms	Fully on-chip	Enabled
[8]	Discrete pulses, CHE injection	500 pA - 1 μA ($\Delta V_{fg} \approx 0.3$ V)	9 bits	-	Only I-V on-chip	Enabled
[114]	Discrete pulses, CHE injection, Differential only	$\Delta V_{fg} = 2$ V	10 bits	50 ms	Off-chip	No
[22]	Continuous time, CHE injection	10-640 pA ($\Delta V_{fg} \approx 0.13$ V)	< 8 bits	-	Off-chip	Enabled
[58]	Pulse-width modulation, Tunneling	$\Delta V_{fg} = 1$ V	6.5 bits	75 μs	Fully on-chip	No

7.5.2 Temperature Dependence

The output voltage of the logarithmic TIA represents the floating-gate voltage and hence is relatively temperature independent even if the floating-gate current varies. Intuitively, the temperature behavior of the logarithmic amplifier is opposite of that of the FGMOS and exactly cancels the change in floating-gate currents. In practice, there is a minor temperature variation due to mismatch in sizes of the actual FGMOS and the feedback transistor in the logamp. In large systems using floating-gates, a floating-gate current reference, such as the one in [101] can be used to bias the array for temperature insensitive currents. The reference itself can be programmed based on floating-gate voltage measurements, which as we mentioned is relatively temperature insensitive.

7.5.3 Programming Time

The time needed to program an FGMOS device comprises the time needed to ramp the voltages (t_{ramp}), injection pulse time (t_{pulse}), measurement time (t_{meas}) and time

to transfer digital bits through SPI (t_{SPI}), all of which gets multiplied by the number of pulses needed to achieve the target (N_{pulses}). So, we can write:

$$T_{prog} = (t_{ramp} + t_{pulse} + t_{meas} + t_{SPI}) \times N_{pulses} \tag{49}$$

For the present implementation, average values of $t_{ramp} + t_{SPI}$ is limited to 0.8 msec. The ramp process is done in small steps to allow the bulk to stabilize and to prevent any chances of latchup. For programming large arrays, the time for ramping voltages up and down would be common to the whole array and its effect on total programming time shall be reduced. The SPI speed is limited by the clock speed of the particular μP used and can be increased by using a different processor. Average values of t_{pulse} and t_{meas} are 0.1 and 0.5 msec respectively. The ADC conversion time can be reduced by employing larger ramp currents and higher clock speeds, the current limitation being set by the 20 MHz processor used. However, the bottleneck for measurement time is not the ADC conversion time but is the settling-time of the log-TIA for smaller currents. The average measurement time, though, can be reduced by using larger ramp currents while measuring larger target currents since the settling time of the log-TIA is reduced in those cases and is no longer the bottleneck. The average value for N_{pulses} is around 35 for the simple programming scheme used. Hence, the average programming speed achievable is around 20 devices/second. It should be noted that the value for N_{pulses} depends on the algorithm used and t_{meas} depends on the accuracy desired (averaging might be necessary). We expect N_{pulses} to reduce to around 25 with a better algorithm.

7.5.4 Future Improvements

In the current implementation, a large fraction of N_{pulses} is needed to come within range of the desired target current. This might be improved dramatically by using a method similar to the one described in [22] for coarse programming. Moreover, we only store one set of injection parameters for the whole array in the LUT. Hence, to

account for mismatch in the parameters, at every step the applied pulse width and V_{ds} value are reduced to a fraction of the actual one needed to achieve the target current. This leads to an increase in the number of required pulses. This can be avoided if characterization parameters are stored for each device in the array, a task that is not prohibitive given the easy availability of digital memory.

The other consideration in using an architecture like this might be the area overhead for the programming infrastructure. If only a few floating-gates are being used, using this whole infrastructure might be prohibitive. In that case, a simpler method might be using off-chip control and measurements. For a production environment, the tester time cost has to be compared with the cost for chip area to make a decision regarding this.

7.6 Conclusions

Though initially the floating-gate device was used as a digital storage, in the recent past, there have been numerous instances of its usage in traditional analog applications such as data converters [20, 50], imagers [7, 29], analog memory [105], offset cancellation in amplifiers [100], low TC current references [92] and many more. This trend requires the accuracy and speed of programming the charge on the gate to increase drastically. A fully integrated architecture for programming floating-gate

Table 7: Table of Parameters

Process	0.35 μm	0.5 μm
Area	250 $\mu m\times$ 1.05 mm	600 $\mu m\times$ 350 μm
Power Supply	3.3V	5V
Injection V_{dd}	$4.8 - 5.6V$	$5.5 - 6.6V$
Resolution of DAC	$7bit$	$7bit$
Switch programming	$N_{rows} \times 100\mu sec$	$N_{rows} \times 100\mu sec$
Bias programming	1 ms/measurement	1 ms/measurement
Dynamic Range	6 pA to 20 μA	NA

based systems with high accuracy, moderate speed and low-power is described in this chapter. It achieves better dynamic range by utilizing a floating-point ADC that has a logarithmic transimpedance amplifier as a first stage followed by a linear ADC. The salient features of the system are presented in Table 7.

Table 6 presents a comparison of this work with other reported implementations. In the table, 'CHE injection' refers to channel hot-electron injection. It should be noted that the accuracy of this implementation increases to around 11 bits with averaging. The errors in modelling injection can be reduced if the dynamic range of operation is restricted to sub-/above-threshold regions, a fact validated by the results in [8]. This implies our implementation can be scaled to 12-13 floating point bits with around 9 bits of mantissa. Programming FGMOS using U-V as described in [17] is not included in the table since it cannot be used to program a wide range of accurate currents.

The most critical piece of this programming architecture is the logarithmic amplifier for measuring a high dynamic range of currents. It was not discussed in details in this chapter since it provides a generic solution to current measurement in sensory or imaging systems and is not limited only to FG programming. I shall discuss this circuit in details in the next chapter.

CHAPTER VIII

MEASURING HIGH DYNAMIC RANGE CURRENTS: ADAPTIVE LOGARITHMIC TRANSIMPEDANCE AMPLIFIER

Transimpedance amplifiers (TIA) are used in a wide variety of applications ranging from microsystem sensors [104] to optical preamplifiers [25, 46, 49, 57, 83, 111]. Challenging work on transimpedance amplifiers involving applications where high speed is required with very low currents has been done in [33, 61]. Optical stimulus localization and centroid computation systems [9, 98] also require sensing low currents spanning several decades. Highly integrated systems demanding low power consumption pose especially difficult challenges. Further complicating the problem is the need for wide dynamic ranges, particularly in sensing systems and reprogrammable systems. Sensing systems interface the physical world which is inherently highly dynamic and reprogrammable systems must cater to a wide variety of applications and specifications. We have fabricated such integrated systems, which serve as the basis for this work. The amplifier topologies presented in this chapter have been used in a floating-gate programming system described in the earlier chapter and also in an imaging system.

Logarithmic compression of the current using a MOSFET in subthreshold has been explored to solve the issue of obtaining a wide dynamic range [33, 61]. For proper phase margin, all of these approaches assume that the poles of the amplifier are far away from the dominant pole set by the feedback element. However, satisfying this assumption entails dissipating considerable power in the amplifier to push the pole away from the maximum input pole set by the highest input current. This simple

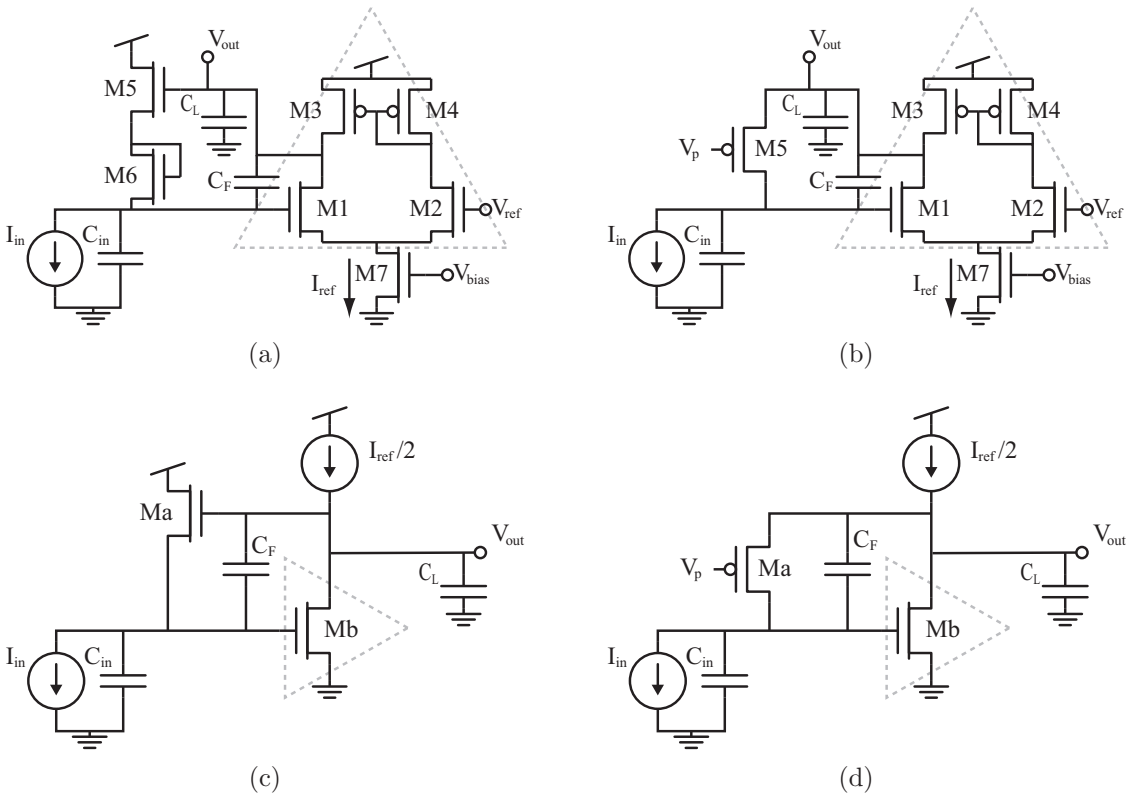


Figure 71: **Topologies of logarithmic TIA**(a) Common-drain logarithmic TIA. (b) Common-gate logarithmic TIA. (c) Simplified version of (a) for small-signal analysis (d) Simplified version of (b) for small-signal analysis

approach is particularly wasteful if it is known that most of the time the input current will be much lower than the maximum value. A solution where the bias current of the amplifier is adjusted has been proposed in [34]. However, the adaptation loop requires a large off-chip capacitor for stability and also degrades the SNR of the system.

The adaptive logarithmic circuit described in this chapter is part of a chip designed for programming floating gates. We analyze in detail the trade-offs involved in a power efficient design of such a system and propose two adaptive strategies to accomplish the same. The two adaptation methods described are very general and can be used in a wide variety of applications depending on specified bandwidth and dynamic range. We show that adapting bias currents is the most power optimal solution for this application. The problem of temperature compensation in these logarithmic amplifiers has been discussed extensively [74] and are not discussed here.

8.1 Overview of Logarithmic Amplifier Design: Problem statement

The fundamental function of a logarithmic transimpedance amplifier is the generation of an output voltage proportional to the log of an input current. The traditional implementation of this logarithmic I-V conversion is done by a passing the current into a diode, a BJT, or a MOSFET operating in its subthreshold, exponential region of operation. In the BJT and subthreshold MOSFET, the current may be passed into a terminal with large exponential conductance (emitter or source), or a node performing a diode connected configuration. The problem with these elements is that the small signal bandwidth at low currents is limited, since the G_{m} of these elements is set by the input current, $G_{\mathrm{m}} = I_{\mathrm{in}}/U_{\mathrm{T}}$, where U_{T} is the thermal voltage equal to kT/q. The approach then becomes one of using an amplifier in a feedback loop, as in Fig. 71, to alter the effective conductance to be AG_{m}, thus increasing the bandwidth by the amplifier gain, A from G_{m}/C to AG_{m}/C. The primary design parameters and corresponding notations that are used in the remaining chapter are listed in Table 8 for convenience.

The final parameter, $I_{\mathrm{in,max}}$ or DR, is relevant because the design is a multiple pole, feedback system, which implies another design specification of stability. The functions to be optimized can be power consumption, noise, and area. In this work we optimize the power consumption. The analysis will show the the basic design of such a structure assuming sub-threshold operation yields an amplifier current consumption given by

$$I_{\mathrm{ref}} > U_{\mathrm{T}}^2 \frac{C_{\mathrm{IN}} C_{\mathrm{L}} (BW^2)(DR)}{I_{\mathrm{in,min}}}. \tag{50}$$

So, the challenges include coping with a power requirement that increases linearly with the required dynamic range and quadratically with bandwidth. It will be shown how adaptive approaches can effectively reduce the power consumption's relation to

Table 8: List of Symbols

BW	Desired bandwidth.
A	Amplifier gain.
I_{ref}	Bias current in the amplifier.
C_{IN}	Capacitance at the input node of the amplifier.
C_{F}	Capacitance in feedback across the amplifier.
C_{L}	Capacitance at the output of the amplifier.
$I_{\text{in,min}}$	Minimum Input Current.
$I_{\text{in,max}}$	Maximum Input Current.
DR	Dynamic range = $\frac{I_{\text{in,max}}}{I_{\text{in,min}}}$.

dynamic range. This is critical in the systems where the desired dynamic range is several orders in magnitude and the reduction is significant.

8.2 Logarithmic Transimpedance Amplifier: Topologies

In this section, we introduce two topologies which operate as wide dynamic range transimpedance amplifiers by log-compressing the input current using the exponential characteristics of MOS transistors in the sub-threshold regime. The small-signal transfer function is derived first, followed by a discussion on minimum power dissipation to meet bandwidth and dynamic range specifications. In the remaining sections of the chapter, it is assumed the desired specifications of the TIA conform with the floating-gate programming application.

8.2.1 Common-drain topology

Fig. 71(a) depicts the structure of a common-drain logarithmic transimpedance amplifier. It should be noted that in our convention we name the transimpedance amplifier by the type of feedback stage employed and not the type of amplifier used. Hence a common-drain logarithmic amplifier has a common-drain or a source-follower feedback stage as in 71(a). Transistors M1 - M4, form a basic differential amplifier with M7 as the current source. M5 and M6 form the feedback element. The operation of the circuit is as follows: the amplifier tries to hold the input node fixed at V_{ref}, thus forcing I_{in} to flow through M5 and M6 and not the capacitor. M5 and M6 form an equivalent transistor with an effective κ, κ_{eff} lower than that of a single transistor, where κ is the inverse of the subthreshold slope factor [106]. The logarithmic conversion of current to voltage is obtained naturally by the exponential I-V relation of a MOS transistor in sub-threshold. The DC output voltage can be expressed as:

$$\begin{aligned} V_{\text{out}} &= \frac{V_{\text{ref}} + (\kappa+1)U_{\text{T}} \ln\left(\frac{I_{\text{in}}}{I_n}\right)}{\kappa^2} \\ &= \frac{V_{\text{ref}}}{\kappa^2} + \frac{U_{\text{T}}}{\kappa_{eff}} ln\left(\frac{I_{\text{in}}}{I_n}\right), \end{aligned} \tag{51}$$

where U_{T} is the thermal voltage and I_n is the pre-exponential factor in the I-V relation of a sub-threshold NMOS.

A version of this circuit was created in 0.5μm CMOS. Fig. 72 shows the measured and simulated current voltage relation with both curves showing good correlation. The slope of the curve changes at high currents because the transistor transitions from below threshold operation to above threshold operation and thus the voltage at the gate increases in proportion to the square root of current instead of the logarithm. A second order curve fit to this plot for currents ranging from $200fA$ to $2\mu A$ gives a linear slope of 0.55, which translates to κ_{eff}=0.47. The ratio of the second order term to the first order term is -45dB.

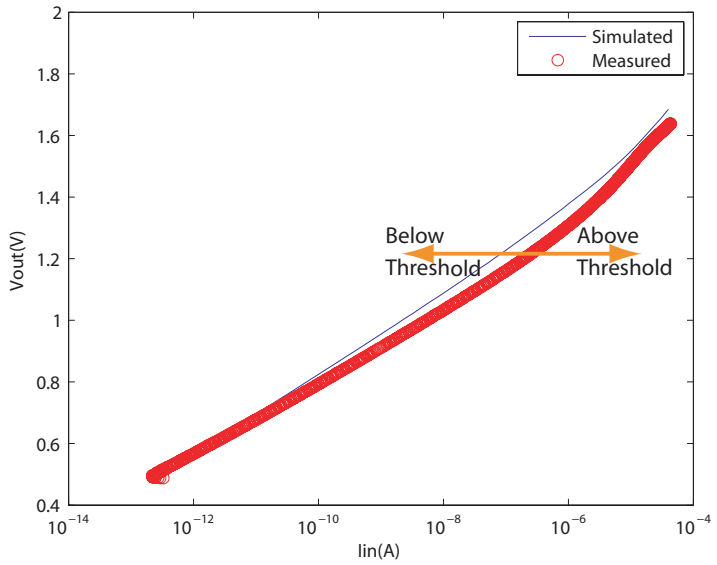


Figure 72: **Current-Voltage characteristics:** Measured and simulated current to voltage curves at DC. At higher currents, the slope becomes steeper because of the MOSFET entering above threshold region from sub-threshold.

8.2.1.1 Small-signal Analysis

To generate a small signal transfer function for the structure, Fig. 71(c) can be used. It is to be noted that Mb in 71(c) represents the amplifier while Ma in 71(c) represents M5 and M6 in Fig. 71(a) and has an effective degenerate G_{ma} given by $\kappa G_{m5}/(\kappa+1)$ which actually encodes the effect of κ_{eff}. This simplification ignores the effect of the current-mirror pole at the gates of M3-M4 as it occurs at much higher frequencies compared with the dominant pole of the amplifier at its output. Also a zero at exactly twice the frequency of the pole is created due to the two paths to the output reducing its effect even more. However, it should be noted that this simplified model is used only for analysis, while simulations are performed on the actual circuit. Analyzing Fig. 71(c), we get:

$$\frac{V_{\rm out}(s)}{I_{\rm in}(s)} \approx \frac{A}{G_{\rm ma}(1+A)} \frac{1-\frac{sC_{\rm F}}{G_{\rm mb}}}{1+as+bs^2}; \qquad (52)$$
$$a = \frac{C_{\rm L}G_{\rm ma} + C_{\rm F}G_{\rm mb} + C_{\rm IN}G_{\rm ob}}{G_{\rm ma}G_{\rm mb}};$$
$$b = \frac{C_{\rm F}(C_{\rm IN}+C_{\rm L}) + C_{\rm IN}C_{\rm L}}{G_{\rm ma}G_{\rm mb}},$$

where $C_{\rm F}$ is the sum of $C_{\rm gdb}$, $C_{\rm gsa}$ and any explicit compensation capacitor placed across Mb, $G_{\rm ob}$ is the output conductance of the amplifier and A is the gain of the amplifier equalling $G_{\rm mb}/G_{\rm ob}$. It should be noted that $C_{\rm gsa}$ in the above equation is the equivalent capacitance due to the series combination of $C_{\rm gs5}$ and $C_{\rm gs6}$ in Fig. 71(a) and is thus equal to half the value of any of them (assuming M5 and M6 are similarly sized). The poles of the circuit are calculated considering two particular cases of interest.

CASE I : $C_{\rm IN} >> C_{\rm F}$

This is the most frequently encountered scenario when the input capacitance dominates the frequency response. Using the dominant pole approximation and noting that $G_{\rm mb} >> G_{\rm ma}$, the poles are given by:

$$p_1 \approx -\frac{AG_{\rm ma}}{C_{\rm IN}} \quad ; \quad p_2 \approx -\frac{G_{\rm ob}}{C_{\rm F}+C_{\rm L}}. \qquad (53)$$

CASE II : $C_{\rm F} >> C_{\rm IN}$

In this case, the input capacitance is so small that generally an explicit $C_{\rm F}$ is needed to robustly design the input pole. In this case:

$$p_1 \approx -\frac{G_{\rm ma}}{C_{\rm F}} \quad ; \quad p_2 \approx -\frac{G_{\rm mb}}{C_{\rm IN}+C_{\rm L}}. \qquad (54)$$

The value of p_1 can be intuitively computed by noting that the total capacitor at input, $C_{\rm tot}$ and the input impedance, $Z_{\rm in}$ are given by:

$$C_{\rm tot} = C_{\rm IN} + AC_{\rm F} = A \times C_{\rm eff} \quad ; \quad Z_{\rm in} \approx \frac{1}{AG_{\rm ma}}, \qquad (55)$$

where A is the gain of the amplifier. Thus the amplifier effectively reduces any input capacitance by the magnitude of its gain. Also, it is important to realize that adding the degeneration transistor does not hamper the bandwidth much since the increase of $Z_{\rm in}$ is almost compensated by the decrease in $C_{\rm F}$.

From the above analysis we see that p_1 depends on the input current and thus moves to higher frequencies as the input current increases.

8.2.1.2 Power Dissipation Limits

In this text, minimizing power dissipation is used synonymously with minimizing $I_{\rm ref}$, the bias current of the amplifier. It is assumed that $I_{\rm ref}$ flows through Mb in Fig. 71 (c) and (d). κ is assumed 1 for simplicity. It is also assumed that typically $C_{\rm IN} >> C_{\rm F}$ unless otherwise mentioned.

The conditions constraining $I_{\rm ref}$ are bandwidth>BW and phase margin>45° for all currents. Since the minimum bandwidth occurs when the input current is minimum, it is sufficient to ensure that the bandwidth at minimum input current is larger than BW. Thus, we need:

$$\frac{AI_{\rm in,min}}{C_{\rm IN}U_{\rm T}} > BW, \tag{56}$$

where it is assumed that $C_{\rm F} < C_{\rm IN}/A$. (56) defines the minimum gain needed to meet the bandwidth specification and enables the designer to choose the number of stages. Here we assume that a one-stage structure is sufficient. The second equation comes from the phase margin specification:

$$p_{2,\rm OPENLOOP} > BW_{\rm max} = \frac{AI_{\rm in,max}}{C_{\rm IN}U_{\rm T}}. \tag{57}$$

This second pole of the system is actually the first pole of the amplifier. Thus the power dissipation of the amplifier can be found by considering it to be proportional to the gain-bandwidth product :

$$I_{ref} > \frac{U_T^2 C_L C_{IN} (BW)^2 DR}{I_{in,min}}. \tag{58}$$

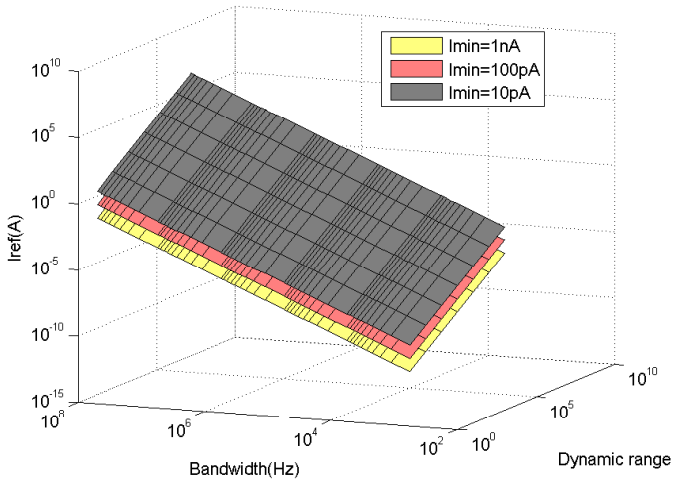


Figure 73: **Power requirement of non-adaptive log-TIA:** Plot showing dependance of power dissipation on dynamic range is linear and on bandwidth is quadratic. Also the required power increases for lower values of $I_{in,min}$ showing dependance of power on speedup.

(58) is a very important equation since it shows the dependance of the power dissipation on the design parameters. The power can be seen to be proportional to dynamic range, input and output capacitances and the square of the bandwidth. The term $I_{\text{in,min}}$ in the denominator shows that the power is actually a function of the desired speedup or $\frac{BWU_TC_{IN}}{I_{\text{in,min}}}$. It should be noted that the term $\frac{I_{\text{in,min}}}{U_TC_{IN}}$ represents the natural bandwidth of the system. Fig. 73 demonstrates the dependance of the power dissipation on bandwidth and dynamic range. This strong dependance can be nullified to a great extent by using adaptation as will be shown later.

8.2.2 Common-gate Topology

Fig. 71(b) depicts the structure of a common-gate logarithmic transimpedance amplifier. Again, a common-gate transimpedance amplifier refers to an amplifier with a common-gate stage in feedback. The amplifier structure is kept similar to the previous case. Only one transistor, M5, is shown as the feedback element, though source degeneration like the earlier case may be used here as well. The operating principle is similar to what has been previously discussed. The logarithmic conversion is obtained from the subthreshold exponential current relationship between current through M5

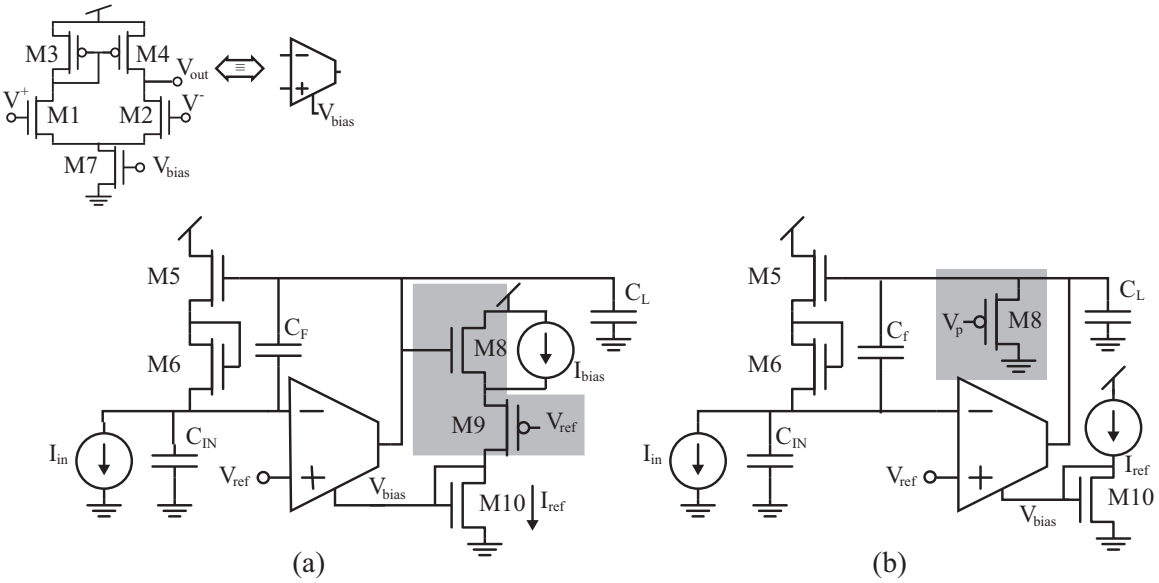


Figure 74: **Topologies of Adaptation:** (a) Circuit for adaptation of amplifier bias current based on input current. The output voltage is taken as a representation of the input current and used to produce I_{adapt}. (b) Second scheme using a PMOS in the adaptive network to reduce the output resistance and hence gain of the amplifier when the input current is large. Thus loop stability is maintained over the current range.

and the source voltage of M5. Thus the logarithmic relationship is limited to the subthreshold region of operation for M5.

$$I = I_{\rm p} e^{(\kappa(Vdd-Vg)-(Vdd-Vout))/U_{\rm T}}$$

$$V_{\rm out} = ln\left(\frac{I}{I_{\rm p}}\right) U_{\rm T} + \kappa V_{\rm g} + (1-\kappa)Vdd, \tag{59}$$

where $U_{\rm T}$ is the thermal voltage, I_p is the pre-exponential factor in the I-V relation of a sub-threshold PMOS and $V_{\rm g}$ is the bias gate voltage.

8.2.2.1 Small-signal Analysis

Fig. 71(d) can be used to generate a small signal transfer function for the common-gate structure. Here again, Mb represents the amplifier while Ma represents the

feedback element. Analyzing Fig. 71(d), we get:

$$\frac{V_{\text{out}}(s)}{I_{\text{in}}(s)} \approx \frac{1}{G_{\text{sa}}} \frac{1 - \frac{sC_{\text{F}}}{G_{\text{mb}}}}{1 + as + bs^2}; \tag{60}$$

$$a = C_{\text{IN}}/G_{\text{mb}} + C_{\text{F}}/G_{\text{sa}} + (C_{\text{IN}} + C_{\text{F}})/AG_{\text{sa}};$$

$$b = \frac{C_{\text{IN}}C_{\text{F}} + C_{\text{L}}C_{\text{IN}} + C_{\text{L}}C_{\text{F}}}{G_{\text{sa}}G_{\text{mb}}},$$

where G_{sa} is the source conductance of Ma and other symbols are as described previously. Here, C_{F} is the sum of the C_{gd} of Mb and any explicit compensation capacitor, with no contribution from Ma as in the common-drain topology. Most amplifiers beyond a single transistor amplifier would make the C_{F} term insignificant. Here we consider only the case $C_{\text{IN}} >> C_{\text{F}}$ as that is representative of the scenario is which this structure is used. Using the dominant pole approximation and noting that $G_{\text{mb}} >> G_{\text{ma}}$, the poles are given by:

$$p_1 \approx -\frac{AG_{\text{sa}}}{C_{\text{IN}}} \quad ; \quad p_2 \approx -\frac{G_{\text{ob}}}{(C_{\text{L}} + C_{\text{F}})}. \tag{61}$$

So, the dominant pole is set by the input node which includes the conductance of the feedback transistor. Again, it is clear that as the input current increases, p_1 increases proportionally.

8.2.2.2 Power Dissipation Limits

The power dissipation constraints on the common-gate structure based on small-signal parameters exactly follows that of its common-drain counterpart. The small signal analysis sets a minimum requirement on I_{b}, the current flowing through the amplifier transistor M_{b}. However, since the input current is sourced from the same current supply as the amplifier, I_{ref} must be larger than the maximum possible input current $I_{\text{in,max}}$, i.e.:

$$I_{ref} = I_{in,max} + I_{b,max}. \tag{62}$$

Hence, the common-drain is always more power-efficient than the common-gate, though this may not be significant if the requirements of the system dictate that

$I_{\mathrm{b,max}}$ need be much larger than $I_{\mathrm{in,max}}$. However, the common-gate topology avoids a fundamental limit of the common-drain configuration. In the common-drain configuration, the $\mathrm{C_{GS}}$ of $\mathrm{M_a}$ is the minimum value of C_{F}, which gets Miller-multiplied by the gain of the amplifier. This inherently sets a limit on the maximum bandwidth attainable (f_{t}) at a particular input current. So if the desired bandwidth is larger than this, the only option is the common-gate topology. With these points in mind we shall focus on the common-drain topology in the remainder of the chapter.

8.3 Adaptation in the Logarithmic Amplifier

In the last section, it was shown that the small-signal poles of the circuit move depending on input current. Two possible ways of designing this circuit to be stable over a wide range of input currents are:

1) For maximum bandwidth at lowest input currents, bias the amplifier by a large current to move p_2 sufficiently higher than the largest possible value of p_1. But this solution obviously entails large power dissipation.

2) For minimum power dissipation, fix a particular bias current and then suitably choose C_{F} such that largest p_1 is sufficiently smaller than p_2. This clearly sacrifices bandwidth.

In this section, we explore two solutions which elegantly trade-off bandwidth and power to meet the specifications. The properties of each adaptation method are discussed first followed by a derivation of its power requirements.

8.3.1 Configuration I: Adaptation of Amplifier Bias Current

The first method senses the input current magnitude and uses it to set the amplifier's bias current. This method is particularly useful for wide dynamic-range systems as will be shown.

8.3.1.1 Operation

Fig. 74(a) depicts the schematic of the bias current adaptation method. Transistors M8 and M9 replicate I_{in} to produce the adaptive current I_{adapt}. M9 acts as a follower to hold the source of M8 constant. In a small-signal sense, the source of M8 sees an impedance of $1/G_{\text{m9}}$. Since the difference between V_{out} and V_{ref} encodes the value of I_{in}, V_{ref} is applied to the gate of M9 to make I_{adapt} directly dependant on I_{in}. V_{ref} sets the voltage across the current source being measured and hence varies with application. The current through M10 is mirrored with a typical gain of K $\approx$ 10. To find an expression for I_{adapt}, we equate the currents through M8 and M9 to obtain:

$$I_8 = I_{\text{n}} e^{(\kappa V_{\text{out}} - V_{\text{S}})/U_{\text{T}}}; I_9 = I_{\text{p}} e^{(\kappa V_{\text{S}} - V\text{ref})/U_{\text{T}}}$$

$$I_{\text{adapt}} = K I_{\text{in}} e^{((1-\kappa)V_{\text{ref}} - \alpha)/((1+\kappa)U_{\text{T}})}, \tag{63}$$

where V_{S} is the voltage at the source of M8, α is given by $U_{\text{T}}\ln(I_{\text{n}}/I_{\text{p}})$ and other symbols are as previously described. The adaptation circuitry moves p_2 by increasing the tail current of the differential amplifier but does not modify V_{out} due to the high Common Mode Rejection Ratio (CMRR) of the amplifier. A current source is added parallel to M8 to ensure that a minimum bias current always flows through the differential pair ensuring a minimum amplifier speed. This helps particularly in step-responses for low currents. The Loop Gain of this adaptation loop is very low due to the high CMRR of the amplifier and thus there is no possibility of instability. Another attractive property of this adaptation is that any noise contributed by the transistors M8 - M11 is rejected by the CMRR. The bandwidth of the adaptation loop is set by the sum of I_{in} and the fixed bias current. In this implementation, p_1 is always the dominant pole. Consequently bandwidth always scales linearly with G_{m5}. G_{m5} varies linearly with I_{in} when M5 is in sub-threshold and as the square root of I_{in} when M5 is above threshold.

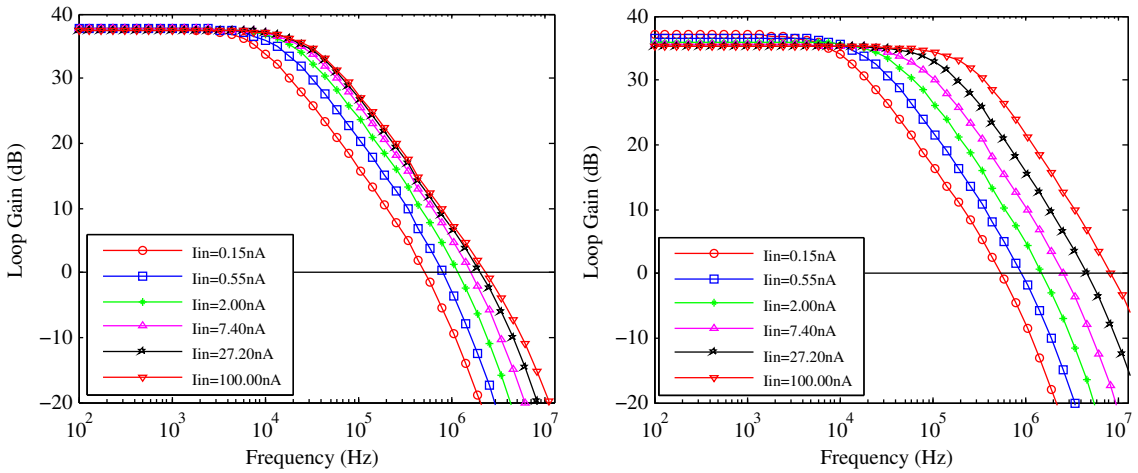


Figure 75: **Loop gain of bias adaptation technique:** (a) Without adaptation the bandwidth does not increase uniformly with current indicating that the output pole becomes dominant for large input currents. (b) With adaptation the loop bandwidth increases with input current indicating that the output pole moves to higher frequencies with increasing input current.

The bandwidth of the circuit can be improved slightly by including cascode transistors in the differential amplifier as it would remove the contribution of C_{gd1} to C_{F}. This scheme of adapting the bias current can be applied to other amplifier structures like a folded cascode amplifier or a standard nine transistor OTA. The simulated plots of the Open Loop gain of the TIA biased at a baseline value of I_{ref}=128nA are shown in Fig. 75. Fig. 75(a) shows that without adaptation loop bandwidth stops increasing with input current when the input current crosses the value of $\approx$ 70nA indicating that the dominant pole switched from input to output. Fig. 75(b) shows the loop bandwidth increasing uniformly with input current indicating that the input pole is always dominant. Fig. 76 shows the measured bandwidth of this configuration with increasing input currents. The slope becomes smaller at higher currents because the feedback transistor moves into above threshold region. From this plot, the value of C_{eff} is extracted to be 22fF.

Fig. 77 demonstrates measured adaptation of the bias current of the amplifier. The total current drawn from the power supply is initially dominated by the baseline

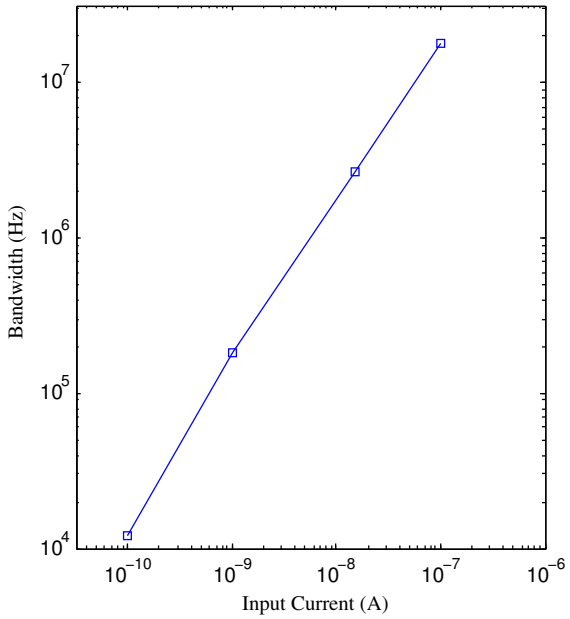


Figure 76: **Variation of bandwidth:** Measured small signal bandwidth based on step responses of configuration I. From this plot, we extract the value of C_{eff} to be 22fF.

value of I_{ref}=128nA when the adaptation current is much smaller than that, but after a certain value of I_{in} the adaptation current dominates the total current. From this curve, the average (geometric mean) power consumed from the 3.3V power supply is 3.45μW. In the next sub-section we discuss the power dissipation limit for this type of adaptation.

8.3.1.2 Power Dissipation

The first constraint based on bandwidth requirement is the same as (56). For the second constraint, we know that I_{ref} in this case is variable (as it is adapting) and relate it to the value of I_{in}:

$$p_{2,\text{OPENLOOP}} \geq \frac{AI_{\text{in}}}{C_{\text{IN}}U_{\text{T}}}, \tag{64}$$

where the symbols used were introduced in the last section on power dissipation. Similar to previous sections, the power is found by considering it to be proportional to gain-bandwidth:

$$I_{\text{ref}} > \frac{U_{\text{T}}^2 C_{\text{L}} C_{\text{IN}} (BW)^2 I_{\text{in}}}{I_{\text{in,min}}^2}. \tag{65}$$

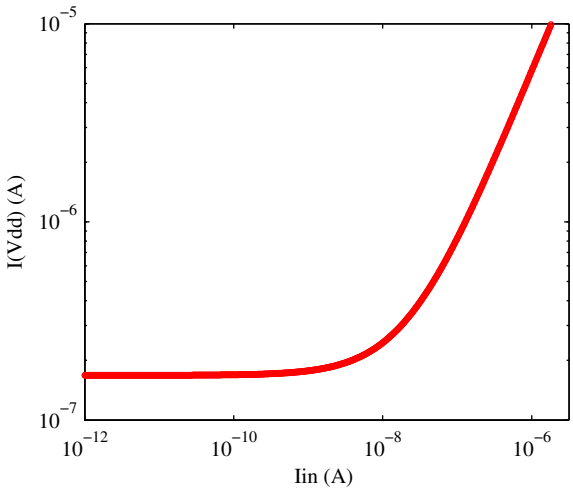


Figure 77: **Bias current adaptation:** Measured current drawn from the power supply demonstrating adaptation of the amplifier bias. The curve is flat initially when the adaptation current is smaller than the baseline value of I_{ref} set at 128nA.

In order to find the average power dissipation it should be noted that due to the nature of the data (varying over decades), the geometric mean is the proper measure of average. Therefore,

$$I_{\text{ref}}(avg) = \frac{U_{\text{T}}^2 C_{\text{L}} C_{\text{IN}} (BW)^2 \sqrt{DR}}{I_{\text{in,min}}}. \tag{66}$$

Thus, comparing (58) and (66) we see that adaptation improves the power dissipation by a factor of the square root of the dynamic range which can be as large as one thousand for a system operating over six decades of current.

8.3.2 Configuration II: Adapting output resistance of the amplifier

The second method reduces the gain of the amplifier at higher input currents when the speedup requirement is typically much lower. It achieves this gain reduction by lowering the impedance at the amplifier's high gain node, which pushes the amplifier's dominant pole to higher frequencies.

8.3.2.1 Operation

The second adaptation method is depicted in Fig. 74 (c). The current in transistor M8 approximately replicates the variations in I_{in}. As I_{in} increases so does V_{out}. Being

the source voltage of M8, an increase in V_{out} increases the current through M8. As its conductance comes in parallel to the amplifier's output conductance, it starts resistively loading the output of the amplifier at high enough I_{in} values. This gain reduction at higher current values allows the feedback loop to be stable even when p_1 approaches p_2. Since the loading depends on the value of V_{p} with respect to V_{out}, V_{p} is chosen from simulations based on the other parameter values. In practice, it is advisable to allow for trimming V_{p} to account for variation in the estimated capacitances which might affect the phase margin at some currents. This method of adaptation assumes that the desired bandwidth does not scale with input current as the lowering of the loop gain means C_{eff} increases at higher currents. The design has to be such that the increase in I_{in} and C_{eff} allow the system to maintain a minimum bandwidth over the entire range of currents. However, this method requires that the amplifier supply the input current directly thus necessitating the amplifier bias current to be larger than the maximum input current. In this sense it is quite similar to the common-gate topology of logarithmic amplifiers and hence dissipates more power for wide dynamic range inputs. The significance of this limitation depends on whether the amplifier already required currents larger than $I_{\text{in,max}}$ to satisfy the BW requirements at $I_{\text{in,min}}$. Fig. 78 demonstrates the effect of gain reduction with increasing input current to maintain stability.

8.3.2.2 Power Dissipation

Following the earlier derivations, we use the constraint that the minimum desired bandwidth is BW to satisfy :

$$\begin{aligned} BW &\leq \frac{G_{\text{ma}}}{C_{\text{F}} + \frac{C_{\text{IN}}}{A}} \approx \frac{I_{\text{in}}}{U_{\text{T}}\frac{C_{\text{IN}}}{A}} \\ \Rightarrow I_{\text{ref}} &\geq \frac{AI_{\text{in}}(I_{\text{in}} + U_{\text{T}}C_{\text{IN}}BW)}{AI_{\text{in}} - U_{\text{T}}C_{\text{IN}}BW} = f(I_{\text{in}}), \end{aligned} \tag{67}$$

where G_{ad} is the source conductance of the adaptation MOS added to the output of the amplifier. So, I_{ref} has to be always larger than $f(I_{\text{in}})$ when I_{in} varies over

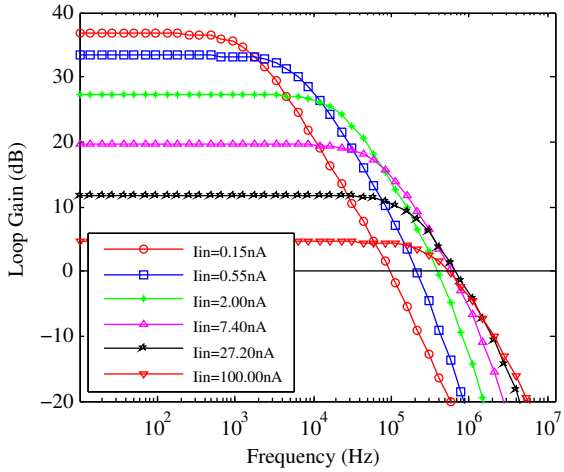


Figure 78: **Loop gain of gain adaptation technique:** The loop-gain for configuration II drops as current increases. This assures stability while ensuring a minimum bandwidth is still obtained.

a specified range. We can make some approximations to get a rough idea about the range of $I_{\rm ref}$ based on this equation. If DR $> 10A$, $I_{\rm ref} > I_{\rm in,max}$ is a sufficient condition. If DR $< A/10$, $I_{\rm ref} > AI_{\rm in,min}$ is the pertinent equation. For the general case, a solution can be graphically obtained by plotting $f(I_{\rm in})$. The second constraint using the phase margin condition can be obtained as:

$$\frac{G_{\rm ad}+G_{\rm o2}}{C_{\rm L}} \geq \frac{AI_{\rm b}I_{\rm in}}{U_{\rm T}C_{\rm IN}(AI_{\rm in}+I_{\rm b})}$$
$$\Rightarrow I_{\rm ref}^2 + I_{\rm ref}I_{\rm in}(2A - \frac{A^2C_{\rm L}}{C_{\rm IN}}) + A^2I_{\rm in}^2(1+\frac{C_{\rm L}}{C_{\rm IN}}) \geq 0. \quad (68)$$

From (68) we can infer the range of $I_{\rm ref}$ under some simplifying assumptions.

CASE I : $AC_{\rm L}/C_{\rm IN} >> 1$

(68) reduces to:

$$I_{\rm ref} \geq A^2 I_{\rm in,max}\frac{C_{\rm L}}{C_{\rm IN}} = \frac{U_{\rm T}^2 C_{\rm L}C_{\rm IN}(BW)^2(DR)}{I_{\rm in,min}}. \quad (69)$$

Comparing this with (66), it is clear that this method consumes more power than adapting $I_{\rm ref}$.

CASE II : $AC_{\rm L}/C_{\rm IN} < 2$

In this case, (68) does not put any constraint on $I_{\rm ref}$. The only constraint is from

(67). Firstly, assuming DR$>10A$, we get:

$$I_{\text{ref}} \;\geq\; I_{\text{in,max}} = \frac{DR}{A}\frac{U_{\text{T}}^2 C_{\text{L}} C_{\text{IN}} (BW)^2}{I_{\text{in,min}}}. \tag{70}$$

Comparing with (66), again we see that this solution consumes more power if $DR > A^2$. This case corresponds to a situation where the desired dynamic range is much more than required speedup.

In the second case, assuming DR$< A/10$, we get:

$$I_{\text{ref}} \;\geq\; AI_{\text{in,min}} = \frac{U_{\text{T}}^2 C_{\text{L}} C_{\text{IN}} (BW)^2}{I_{\text{in,min}}}. \tag{71}$$

Thus in this case, comparing with (66), output impedance adaptation consumes lesser power than I_{ref} adaptation by a factor of $\approx \sqrt{DR}$. From (70) and (69), for a floating-gate programming system that requires measurements spanning a very wide range of currents (DR$\approx 10^6$) and a modest speedup (A≈ 100), adapting the amplifier's bias current is the better solution compared to adapting the output impedance.

8.4 Noise Performance of the Adaptive Logarithmic TIA

In this section, the noise performance of the circuit in Fig. 74(a) referred to as configuration I is discussed in detail. Configuration II can be analyzed following the analysis presented.

For the noise analysis, we consider channel noise current sources for each transistor separately and compute their contribution to voltage noise power, $\hat{V}_{\text{out}}^2$ at the output. The total noise power is found by adding the noise power due to these uncorrelated sources. The calculation can be simplified by noting that noise currents due to transistors M7 - M11 appear as a common-mode signal to the differential amplifier and are thus attenuated by the CMRR. Also, M5 and M6 are in saturation with equal values of G_{m} denoted by G_{m5} in the following analysis. Then we have :

$$\hat{V}_{\text{out}}^2 \approx \frac{(\tilde{i}_5^2 + \tilde{i}_6^2)}{G_{\text{m5}}^2} + \frac{(\tilde{i}_1^2 + \tilde{i}_2^2 + \tilde{i}_3^2 + \tilde{i}_4^2)}{G_{\text{m1}}^2}, \tag{72}$$

where $\tilde{i}_k^2$ denotes the channel current noise power of transistor Mk. The output noise can also be referred back to the input as a current noise whose value is given by:

$$\hat{i}_{in}^2 \approx \left(\frac{\kappa}{\kappa+1}\right)^2 (\tilde{i}_5^2 + \tilde{i}_6^2) + \left(\frac{\kappa}{\kappa+1}\frac{G_{\mathrm{m5}}}{G_{\mathrm{m1}}}\right)^2 (\sum_{k=1}^{4} \tilde{i}_k^2) \tag{73}$$

From (73) it is evident that as the adaptation makes $G_{\mathrm{m1}} >> G_{\mathrm{m5}}$ the noise due to the amplifier becomes negligible. The noise at the output is going to have a thermal noise component and a component due to 1/f noise. This expression needs to be integrated over the bandwidth of the TIA to get the total integrated output noise. For the analytic derivation, we only consider the thermal noise component of a transistor modelled following [90] as:

$$\frac{\tilde{i}^2}{\Delta f} = 2qI, \tag{74}$$

where q is the electronic charge and I is the current flowing through the transistor. Using the transfer function from I_{in} to V_{out} calculated earlier, we get the total noise to be:

$$\hat{V}_{\mathrm{out,total}}^2 = \frac{4qI}{G_{\mathrm{m5}}^2} \int \frac{df}{1+\frac{\omega^2}{p_1^2}} = \frac{kT}{(\kappa+1)C_{\mathrm{eff}}} \tag{75}$$

Thus the total integrated noise is independent of I, a result that is expected because the bandwidth increases with I while the voltage noise density at the output decreases with I. An intuitive explanation of this phenomenon based on equipartition of energy is found in [33].

The output voltage noise spectrum up yo 100kHz was measured for configuration I using a spectrum analyzer(Stanford research System's SR 780) for different input currents; the result is plotted in Fig. 79. The bandwidth obtained from the noise plots corroborate the value of C_{eff}≈20fF which is close to the value calculated based on process parameter values in the AMI 0.5μm process and also from measured step responses. From the figure, integrating the noise spectrum gives a total noise of 700μV rms. The theoretical equation for thermal noise predicts a value of 396μV rms with the

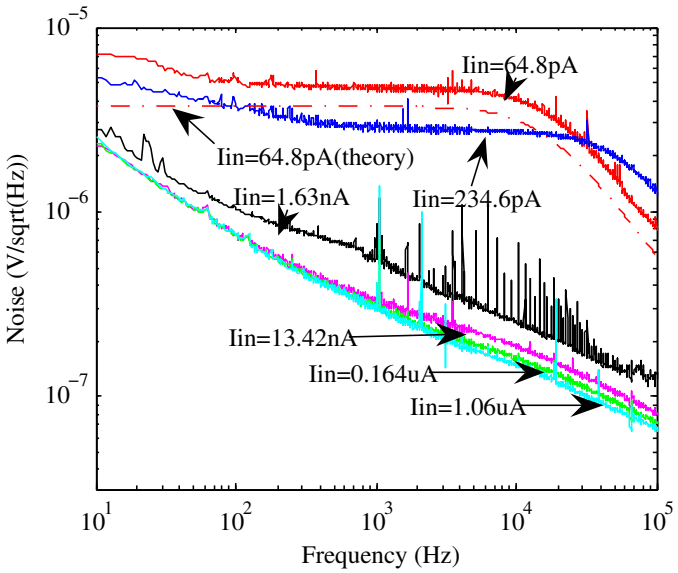


Figure 79: **Noise performance:** Measured output voltage noise spectrum for Configuration I. The thermal noise floor reduces and the corner frequency increases with increasing input current. At higher currents the 1/f noise dominates in a 5kHz band.

difference probably being due to 1/f noise and measurement noise. The theoretically calculated thermal noise density for $I_{\text{in}} = 65$ pA is 3.7μVrms which matches closely with the measured noise density of 4.7μV rms as shown in Fig. 79. Also, the fact that the adaptation circuitry does not indeed add more noise was verified by comparing the noise spectra of a circuit in Configuration I with a similar circuit but with no adaptation circuit. Both the circuits were biased at an input current $\approx$ 1nA. The results are plotted in Fig. 80. The SNR can now be computed using the transfer function as follows:

$$SNR_{\text{power}} = \frac{(\kappa+1)^3}{\kappa^4}\frac{C_{\text{eff}}U_{\text{T}}}{q} \tag{76}$$

Using the earlier obtained value of C_{eff} and choosing κ=1, we get an approximate value of SNR as 45dB. But it should be noted that in applications like floating-gate programming or imaging, the scaling of bandwidth with input current is not required. Rather, this system demands a constant bandwidth of a few kHz depending on a fixed sample rate of the system clock. Thus using a filter after the TIA to limit the bandwidth to say 5 kHz, we get the SNR to scale with I_{in}. The SNR in the case

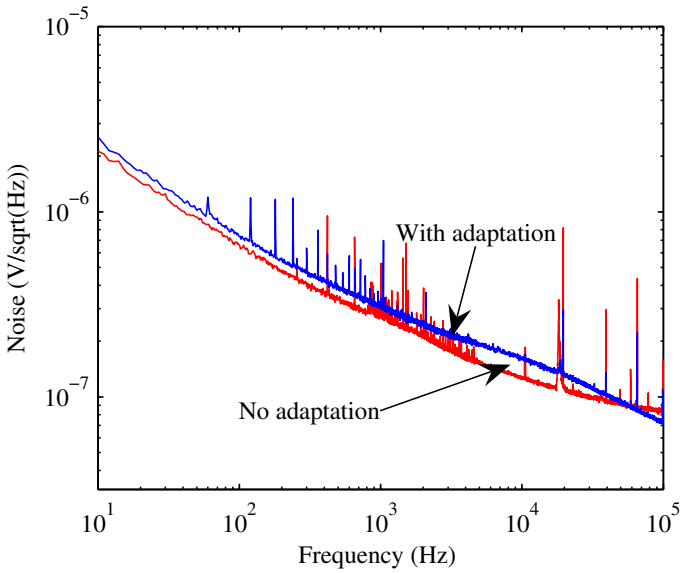


Figure 80: **Adaptation noise:** Comparison of measured noise performance of a circuit with and without adaptation. The adaptation introduces minimal noise

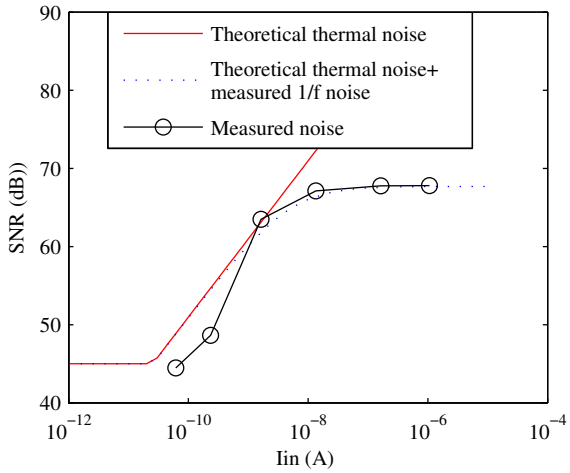


Figure 81: **Signal to noise ratio:** Scaling of SNR with input current for a fixed bandwidth of 5kHz. The plot based on theoretical thermal noise and measured 1/f noise is close to the measured curve.

where the bandwidth is limited to"f_{BW}" is given by:

$$SNR_{power} \approx \left(\frac{\kappa+1}{\kappa}\right)^2 \frac{I_{in}}{qf_{BW}} \tag{77}$$

This equation ignores the noise introduced by the filter itself.

Fig. 81 depicts the theoretically predicted and the measured variation of SNR. Thus, over the plotted range of input currents the average SNR is approximately 65dB. The theoretical curve flattens out at low currents when the bandwidth of the

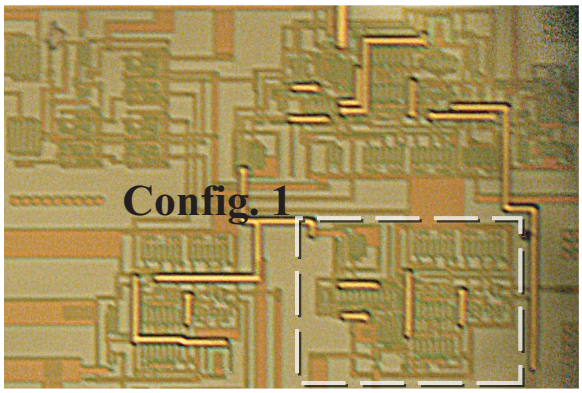


Figure 82: **Die micrograph:** Die Photo of the fabricated chip in 0.5um AMI CMOS process. Configuration I occupies $91 \times 75\mu m^2$ of area.

TIA is lower than 5kHz. It should be noted that the increase in SNR with increasing current saturates for the measured case because the contribution of $1/f$ noise starts dominating in the 5kHz band. A curve plotted with theoretical thermal noise and measured $1/f$ noise corresponds very closely to actual data proving the claim. Fig. 82 shows the micrograph of the fabricated chip.

8.5 Conclusions

Traditionally, different architectures for transimpedance amplifiers including common-source, common-gate and common-drain have been explored. In all these systems, there is an inherent trade-off between input-current noise and bandwidth due to the fixed feedback impedance [28]. For this reason, applications requiring very wide dynamic range and low-noise need some form of gain-adaptation. Techniques that extend the dynamic range right at the preamplifier include varying the preamplifier gain [49, 57, 71], placing a variable signal attenuator before the preamplifier [88, 111] or using two feedback paths with different gains [46]. The first two techniques use neither continuous nor automatic adaptation, but instead have a gain control input which is set by the user. The third technique has the disadvantage of having two outputs which need to be combined using additional circuitry. Table 9 compares this work with some of the references.

Table 9: Comparison of Performance of wide dynamic range TIA designs

Reference	[34]	[49]	[46]	This Work
Process	1.6μm	1.5μm	Discrete	0.5μm
Area	-	$2.9\times3.7mm^2$	-	$91 \times 75\mu m^2$
Supply	-	$\pm$6V	-	3.3 V
DR	100dB	114dB	80dB	140dB
Bandwidth	>10Hz	1kHz	25kHz	>3.5kHz
Power	10pA-1μA	30mW	-	.1μW - 33μW

In this chapter, we analyze in detail the power dissipation for the two main logarithmic transimpedance amplifier topologies and show their functional dependance on speedup ($\frac{BWU_TC_{IN}}{I_{\text{in,min}}}$), dynamic range ($I_{in,max}/I_{in,min}$) and input, output capacitances. We present two adaptation methods to overcome this problem. The first technique adapts the amplifier's bias current depending on input current to maintain bandwidth and stability. The second method reduces the gain of the amplifier at higher currents when the required speedup is less. It is shown that configuration I is ideal for wide dynamic ranges. For this configuration, bandwidth or SNR can be made to scale with input currents. It is also shown that adaptation does not adversely affect bandwidth or noise performance. Measured results show operation over 7 decades of current with an average power consumption of 3.45μW. The average SNR for operation at 5kHz is 65 dB. The adaptation schemes introduced are very general and can be applied to a wide class of circuits.

CHAPTER IX

CONCLUSION

In the earlier chapters of this thesis, I have discussed methodologies for designing compact and low-power analog models of different neurons and the architecture of a reconfigurable processor for neural simulations and analog signal processing. In this chapter, I shall summarize the key points of the research completed and point out several possible future directions.

9.1 Research Summary

In the context of neural systems, this work has introduced the concept of bifurcation based analog neuron design which has resulted in the lowest power dissipating type I neuron reported so far. Chapters II and III discussed the dynamical properties of two neurons exhibiting two different codimension one bifurcations - Hopf and saddle-node. In chapter II, a silicon neuron is modeled and it is demonstrated that the neuron exhibits Hodgkin-Huxley type transient dynamics. In particular, we show a subcritical Hopf bifurcation which is the trademark of Class 2 neural excitability. Also, we demonstrate a bifurcation mechanism involving subcritical Hopf bifurcation and a fold limit-cycle bifurcation that models the excitation block phenomenon. This work enables finding the proper biasing regime for this circuit, a non-trivial task because of the large dimensionality of the parameter space. Also, this is the first low-power on-chip implementation of the circuit as [37] mentions using large bias currents and off-chip capacitors.

In chapter III, we have presented a bifurcation based type I silicon neuron that consumes 1.74 nW at an average spiking rate of 100 Hz, the lowest power among all designs reported till date. Since it is based on a saddle-node bifurcation, it possesses

all the phenomenon/properties associates with the bifurcation. The proposed design mimics the dynamics of a type-I membrane and exhibits refractory period, positive feedback and spike frequency adaptation. We also propose a circuit to extract the nullclines for the system which leads to an algorithm for correctly biasing the circuit. Combined with the ability of floating-gate transistors to set bias voltages locally, this method should reduce variation in silicon neurons while not incurring a significant area penalty.

In general, I have shown that the *power efficiency* in analog models of spiking neural networks, while being far lesser than its biological counterparts, is around ten thousand times better than a digital simulation. This allows us to scale neural simulations to larger networks without running into constraints due to excessive power dissipation. However, communicating spikes across chips will incur an extra power penalty that I have not yet considered. To keep that penalty as low as possible, the strategy should be to hard-wire the dense local interconnections while reserving spike transmission only for sparse long distance connections.

To use these analog circuits instead of a digital simulation, the analog system needs to have the flexibility of reconfiguring the network topology and tuning the neuron parameters. Chapter IV presents one such reconfigurable analog architecture based on the concept of a floating-gate (FG) based field programmable analog array. The improved features of this chip compared to earlier FPAA designs are discussed. The general structure of the chip is composed of an array of computational analog blocks (CABs) within a programmable switch matrix. By modifying the components in the CABs, different FPAAs can be made for different applications. Chapters V and VI present two such different FPAA chips for neural simulations and analog signal processing.

In chapter V, we presented a reconfigurable integrated circuit for accurately describing neural dynamics and computations. Both the topology of the networks as

well as the parameters of the individual blocks can be modified using floating-gate transistors. Neuron models of complexity varying from integrate and fire to Hodgkin-Huxley can be implemented. Computational area efficiency is considerably improved by implementing synaptic weight on the analog switch matrix resulting in all to all connectivity of neurons. We demonstrate dynamics of integrate and fire neurons, Hopf neurons of Hodgkin-Huxley type, inhibitory and excitatory synapses, dendritic cables and central pattern generators.

In chapter VI, the RASP 2.8 IC is described that provides a powerful platform for prototyping and implementing large-scale signal processing applications. The programmable switch matrix composed of floating gate devices shows excellent isolation and can be readily utilized in computation. Different levels of routing allow implementation of high performance circuits while allowing for fast turn-around times. The chip we present is the largest designs reported with around fifty-thousand programmable analog parameters and has significantly more variety in CAB components compared to others.

One of the most important improvements of this family of FPAA chips over its predecessors is the integration of floating-gate programming circuits on the same chip as the FG array. This is enabled by digital-analog converters for supplying gate and drain voltages and a floating-point analog-digital converter that measures the programmed current in the FG device. This has led to improved speed, accuracy and dynamic range of FG programming. The associated circuits and infrastructure are detailed in chapter VII. The central element of the programming infrastructure is a logarithmic transimpedance amplifier that can measure a wide dynamic range of current but dissipates lower power than other designs by employing adaptation. This problem is a general one faced in sensing applications too and hence this circuit is detailed separately in chapter VIII.

This work has laid the foundation for using reconfigurable analog systems for a

wide variety of applications. In the context of analog integrated circuit design, the advent of these reconfigurable chips should usher in a revolution similar to or bigger than what happened in the digital market because of the introduction of FPGAs. The analog FPAAs can be used for prototyping different analog algorithms leading to a reduced number of cycles for the final custom implementation. For applications where the performance on this platform is allowable for the final system, this FPAA chip can be used for the actual product leading to a time-to-market of less than an year. Users should now be able to concentrate more on the system design and signal processing aspects rather than be concerned about the details of every transistor.

9.2 The Future

The work done in this thesis allows one to think about solving problems in a wide variety of fields. I shall elaborate a few topics that I plan to pursue in the following years.

9.2.1 Computational Neuroscience

There are several neuroscience problems that can be studied using these integrated circuits including the ones mentioned below:

1. Differences in computational properties of Hopf and saddle-node neurons
2. Importance of the inhibition time-constant variation in Hodgkin-Huxley models
3. Computational properties of active dendrites and their similarities with hidden markov models
4. Synchronization properties of bursting neurons with different bifurcations resulting in the bursts
5. Effect of noise on synchronization, effect of noise on network properties etc.

While digital simulations will still be popular in the near future for its ease of use and well controlled, accurate simulation engine, I can envision a future where neuroscientists will resort to using these analog chips as a co-processor or for their full simulation. There is still a long road to be traveled before achieving that goal, but the path seems to be clear.

9.2.2 Electrophysiology

The current electronic equipment needed for electrophysiology experiments, in particular, voltage or patch clamps, current clamps and dynamic clamps consist of a bulky instruments mounted on a large rack. The setup is both costly and cumbersome. It also has its limitations, particularly for dynamic clamps, where the delay in digitizing the acquired voltage and computing the feedback current may cause instabilities [84]. Also, the number of channels and maximum update rates have been limited to four and 50 kHz respectively. An FPAA system combining signal processing and silicon ion channel elements can replace this whole lot of equipment and also provide superior performance. The signal processing and conditioning circuits are needed mostly to cancel the parasitic effects of probe resistance and capacitance. The neural elements can be used in dynamic clamps as conductance inputs to the real cell. Since every channel operates independent of any other, the number of channels is limited to only chip area and can easily be as high as sixteen in a 3mm×3mm IC in 0.35 μm CMOS. Also, instability problems should be reduced due to the removal of the analog-digtal converter(ADC) from the loop. This small FPAA board should make electrophysiology setups relatively inexpensive and hopefully make it easier to teach.

9.2.3 Sensor Interfacing

The reconfigurable analog processors that I have developed can be used for several applications other than neural modeling. One such instance is that of a 'smart sensor' or a sensor with an integrated processor. This work is motivated by collaborative

research that I have performed with the MIST laboratory advised by Dr. Levent Degertekin. For the sake of completeness, I shall briefly discuss this next.

I have worked on developing low-noise interface circuits for capacitive micro-machined ultrasound sensors (cMUT) [80] [79] [81] with Dr. S. Peng, Dr. M. Qureshi and Mr. G. Gurun. Our approach of using a capacitive voltage amplifier for sensing changes in capacitance is one of the lowest noise approaches since it eliminates the noise from the feedback resistor in typical transimpedance approaches. The floating-node at the input is stabilized either by a MOS switch, or MOS pseudo-resistor or continuous time injection and tunneling mechanisms. For ultrasound applications, since the current through the cMUT carries the information, using a capacitive amplifier requires a derivative operation down the processing chain. To alleviate this, we have also developed transimpedance approaches to directly transduce the current to a voltage. We have also worked on integrating the MEMS sensor part on the same wafer as the CMOS circuits. This approach reduces parasitic capacitances by an order of magnitude. Initial results of this system are very promising.

I have also worked with Dr. S. Qureshi in designing interface circuits for an optical hearing-aid. The system uses pulses of light shined on a movable optical grating to sense the motion of the grating from the resulting interference pattern. The motion of the grating depends on the incident sound wave. Two approaches have been used to sense the current from the photodiode detector - one is continuous while the other is discrete-time. An analog receiver chain with a passive transimpedance, bandpass filter with gain, peak detector and low-pass filter has been fabricated. The reason for using a passive transimpedance stage is that the speed required is not very high. The chip is designed in $0.35\mu m$ CMOS and operates from a low power supply voltage of 1.5V. Special design techniques are used for low-voltage design. Different peak-detector topologies have been tried to overcome the problems associated with a fixed discharge rate that is most suitable for a fixed modulating frequency. The other

receiver takes advantage of the fact that the received signal is already sampled. So a current-mode delta sigma converter is used to digitize the current directly. Wide input linear-range OTA circuits employing capacitive attenuation are used for the integrators. These circuits have the advantage that the G_m-degeneration circuit does not add noise.

An obvious next step is combining these low-noise sensor interfaces with the reconfigurable processors described earlier. Such an initial design has been developed and described in [78]. The first row of CABs is a special one with high performance LNA circuits that are connected to the pads with minimum switches in the signal path. The amplified signal can be processed by the signal processing circuits in the remaining CABs. The resulting smart sensor can be used for several applications like dynamic clamps in experimental neuroscience, beam-forming in intra vascular ultrasound, hearing aids etc.

9.2.4 Adaptive FPAA - Spike Timing Dependent Plasticity

The FPAA designs described till now, though reconfigurable, are pre-programmed. In other words, the configuration or the parameter values are preset before it is used for an application; it is not dynamically tunable. Using a separate digital controller, the output of the circuits implemented on the FPAA can be monitored to generate the next set of parameters to be programmed on the chip. But this requires a digital controller and the update time may not be well controlled. Instead, it is more useful if the parameters of the circuit could be modified naturally in an analog way based on some system output. This belongs to a class of circuits popularly called 'self-healing' since it can modify or heal its parameters to maintain the output of the system at the desired value. Hence, the system becomes largely immune to drift of parameters due to temperature, ageing etc.

In the context of neural systems, this concept of self-healing coincides with the

concept of 'learning'. There has been several different algorithms for learning in networks of neurons with the most popular ones arguably being back-propagation (in traditional static neural networks) and STDP or spike timing dependent plasticity (in spiking neural networks). A set of new FPAA designs are being developed which allow for modification of FG charge in run-time to explore different learning algorithms. A more custom chip will be made with a dense synaptic matrix with STDP being the supported learning format. Using these two chips, I plan to conduct experiments to understand how the dynamics of individual networks affect the learning properties of the network. Also a neuron simulator needs to be developed to simulate the effect of different hardware learning rules on the learning properties of the network.

In the context of VLSI systems, a slightly different and less flexible format of learning might be more useful. Using the concept of built in self test (BIST) from digital systems, we can fabricate an FPAA which can allow blocks of the chip to be selectively put into 'PROG' mode for testing the properties of its components. The test signals for this block may be obtained from other blocks which are in 'RUN' or operational mode. Then that block can be put in 'RUN' mode with the desired biases based on the tests performed earlier.

Thus, this work seems to have opened several avenues of exciting research over the next few years. Once this programmable analog technology becomes more popular, I believe we will be faced with another problem - that of educating researchers (and learning ourselves) how to efficiently utilize the available resources in this kind of a platform. The design philosophy in this system is going to be different from that of a custom chip since the building blocks and their quantities will differ. Educating other researchers about this technology has also been a part of the work that I have done; I have taught in several workshops and classes but that has definitely not been a major focus yet. Over the years, I envision more effort being spent in teaching people till a day comes when less effort is spent in developing new designs than teaching others to

do system-design in reconfigurable ICs. When we do reach that day, I shall consider the work done in this thesis to have achieved its goal.

APPENDIX A

NORMAL FORM FOR HOPF NEURON

In this section, the derivations of the center manifold reduction and associated normal form described in chapter 2 is presented.

Let us consider a general problem defined by the equation

$$\dot{x} = V(\mu, x), x \epsilon R^4, \mu \epsilon R \tag{78}$$

where μ is a parameter (I_{in} in the present case) and without loss of generality we consider the system to have an equilibrium at the origin for $\mu = 0$. The Taylor expansion of this system around the origin gives rise to a linearized flow given by:

$$\dot{x} = DV(0,0)x, D_x V(\mu, 0)_{ij} = DV(\mu, 0)_{ij} \equiv \frac{\partial V_i}{\partial x_j}(\mu, 0) \tag{79}$$

Let E_s, E_u and E_c denote the linear subspaces spanned by the eigenvectors of DV which have eigenvalues with negative, positive and zero real parts respectively. Let the dimensions of the respective spaces be n_s, n_u and n_c. The non-linear system has stable, unstable and center manifolds denoted by W_s, W_u and W_c which are tangential to the corresponding linear spaces at the origin(Center-Manifold Theorem) [44]. At a Hopf bifurcation, two eigen-values lie on the imaginary axis and hence $n_s = 2$. Since all bounded solutions in the neighborhood of the equilibrium are confined to the center manifold [31], we only need to consider projections of the flow onto this manifold for observing periodic solutions. In order to capture the behavior of the solutions as the parameter is varied, another dummy equation $\dot{\mu} = 0$ is appended to 78 which makes $n_s = 3$ and $x = (x, \mu)$.

Considering $\mu = 0$ to be the critical value we can write (78) as:

$$\dot{x} = DV(0,0).x + N(x), \tag{80}$$

where N(x) denotes the non-linear terms. We can choose variables $x1\epsilon E^c$ and $x2\epsilon E^s \bigoplus E^u$ such that x=(x1,x2) and (80) becomes:

$$\dot{x1} = A.x1 + N_1(x1, x2)$$
$$\dot{x2} = B.x2 + N_2(x1, x2), \tag{81}$$

where A is an $n_c \times n_c$ matrix with all eigenvalues on the imaginary axis, B is an $(n_s + n_u) \times (n_s + n_u)$ matrix with all eigenvalues off the imaginary axis, and N_1 and N_2 are the resulting non-linear terms. The crucial observation in this case is that as W_c is tangent to E^c at x=0 and passes through x=0, we can describe W^c as the graph of a function h(x1) near x=0,

$$h : E^c \longrightarrow E^s \bigoplus E^u, h(x1) = x2 \tag{82}$$

Thus for a trajectory belonging to W^c we can write:

$$x2(t) = h(x1(t)) \tag{83}$$
$$\Rightarrow \frac{dx2}{dt} = [D_{x1}h(x1)].\frac{dx1}{dt} \tag{84}$$
$$= [D_{x1}h].[A.x1 + N1(x1, h(x1))] \tag{85}$$

Now, combining (83) and (81), we have:

$$[D_{x1}h].[A.x1 + N1(x1, h(x1))] = B.h(x1) + N_2(x1, h(x1)) \tag{86}$$

This partial differential equation gives an expression for the center manifold which is solved by considering the Taylor series for the function h near x=0.

Following the above procedure for Hopf bifurcation, we get a reduced two dimensional equation depending on μ. These may be further simplified by near identity co-ordinate changes. As the normal form equations for Hopf is two dimensional, it is useful to cast the equations in the complex form. Let the equation obtained after center manifold reduction be

$$\dot{z} = \lambda z + h(z, \overline{z}); z, \lambda \epsilon C^1 \tag{87}$$

Let the normal form variable be w. Then we know the desired normal form for Hopf is given by:

$$\dot{w} = \lambda w + c_1 w^2 \overline{w} + O(|w|^5) \tag{88}$$

where the cubic coefficient c_1 needs to be determined. To remove the second order term, we consider a change of variables:

$$\begin{aligned} z &= w + \psi(w, \overline{w}) \\ &= w + \psi_{ww}\frac{w^2}{2} + \psi_{w\overline{w}} w\overline{w} + \psi_{\overline{w}\overline{w}}\frac{\overline{w}^2}{2} \end{aligned} \tag{89}$$

Substituting (89) in (87) and equating coefficients we can solve for the function ψ. Using this function, the cubic coefficient can now be obtained. The resulting equation after these simplifications is stated in (11) after converting the variables to polar form.

The same theory can be applied to the other Hopf bifurcation occurring at lower currents leading to the following normal form:

$$\begin{aligned} \dot{r} &= r(0.113\mu - 0.03\mu^2 + (0.61 + 0.061\mu - 0.012\mu^2)r^2) \\ \dot{\theta} &= 2.42 + 0.5\mu - 0.02\mu^2 + (-0.015 - 0.034\mu + 0.014\mu^2)r^2, \end{aligned} \tag{90}$$

REFERENCES

[1] Ahuja, B., Vu, H., Laber, C., and Owen, W., "A Very High Precision 500-nA CMOS Floating-gate Analog Voltage Reference," *IEEE Journal of Solid-State Circuits*, vol. 40, pp. 2364–72, Dec 2005.

[2] Alvado, L., Tomas, J., Masson, S. R.-L., and Douence, V., "Design of an Analogue ASIC using Subthreshold CMOS Transistors to Model Biological Neurons," in *Proceedings of the IEEE Custom Integrated Circuits Conference*, pp. 97–100, May 2001.

[3] Anadigm, "AN120E04 Datasheet configurable FPAA," 2003.

[4] Arena, P., Fortuna, L., Frasca, M., and Patané, L., "A CNN-Based Chip for Robot Locomotion Control," *IEEE Transactions on Circuits and Systems I*, vol. 52, pp. 1862–71, Sept 2005.

[5] Arthur, J. and Boahen, K., "Synchrony in Silicon: The Gamma Rhythm," *IEEE Transactions on Neural Networks*, vol. 18, pp. 1815–1825, Nov 2007.

[6] Babanezhad, J. and Temes, G., "A 20-V four-quadrant CMOS analog multiplier," *IEEE Journal of Solid-State Circuits*, vol. 20, pp. 1158–68, Dec 1985.

[7] Bandyopadhyay, A., Hasler, P., and Anderson, D., "A CMOS floating-gate matrix transform imager," *IEEE Sensors Journal*, vol. 5, pp. 455–62, June 2005.

[8] Bandyopadhyay, A., Serrano, G., and Hasler, P., "Adaptive Algorithm Using Hot-Electron Injection for Programming Analog Computational Memory Elements Within 0.2% of Accuracy Over 3.5 Decades," *IEEE Journal of Solid-State Circuits*, vol. 41, pp. 2107–14, Sept 2006.

[9] Bashyam, A., Furth, P., and Giles, M., "A high speed centroid computation circuit in analog VLSI," in *Proceedings of the International Symposium on Circuits and Systems*, vol. 4, pp. 948–951, May 2004.

[10] Baskaya, F., Reddy, S., Lim, S. K., and Anderson, D. V., "Placement for Large-Scale Floating-gate Field Programmable Analog Arrays," *IEEE Transactions on VLSI*, vol. 14, no. 8, pp. 906–910, 2006.

[11] Basu, A. and Hasler, P., "A Fully Integrated Architecture for Fast Programming of Floating Gates," in *Proceedings of the International Symposium on Circuits and Systems*, pp. 957–60, May 2007.

[12] Basu, A., Odame, K., and Hasler, P., "Dynamics of a Logarithmic Transimpedance Amplifier," in *Proceedings of the International Symposium on Circuits and Systems*, pp. 1673–76, May 2007.

[13] Basu, A., Petre, C., and Hasler, P., "Bifurcations in a Silicon Neuron," in *Proceedings of the International Symposium on Circuits and Systems*, May 2008.

[14] Basu, A., Robucci, R., and Hasler, P., " A Low-Power, Compact, Adaptive Logarithmic Transimpedance Amplifier Operating Over Seven Decades of Current," *IEEE Transactions on Circuits and Systems I*, vol. 49, pp. 2167–77, Oct 2007.

[15] Basu, A., Twigg, C., Brink, S., and et al, P. H., "RASP 2.8: A New Generation of Floating-gate based Field Programmable Analog Array," in *Proceedings of the IEEE Custom Integrated Circuits Conference*, Sept 2008.

[16] Becker, J., Henrici, F., Trendelenburg, S., Ortmanns, M., and Manoli, Y., "A Field-Programmable Analog Array of 55 Digitally Tunable OTAs in a Hexagonal Lattice," *IEEE Journal of Solid-State Circuits*, vol. 43, pp. 2759–68, Dec 2008.

[17] Berg, Y., Wisland, D., and Lande, T., "Ultra Low-Voltage/Low-Power Digital Floating-gate Circuits," *IEEE Transactions on Circuits and Systems II*, vol. 46, pp. 930–36, July 1999.

[18] B.Linares-Barranco, Sanchez-Sinencio, E., Rodriguez-Vazquez, A., and Huertas, J., "A CMOS Implementation of Fitzhugh-Nagumo Neuron Model," *IEEE Journal of Solid-State Circuits*, vol. 26, pp. 956–965, July 1991.

[19] Boahen, K., "Point-to-Point Connectivity Between Neuromorphic Chips Using Address Events," *IEEE Transactions on Circuits and Systems II*, vol. 47, pp. 416–34, May 2000.

[20] Brady, P. and Hasler, P., "Offset compensation in flash ADCs using floating-gate circuits," in *Proceedings of the International Symposium on Circuits and Systems*, vol. 6, pp. 6154–57, May 2005.

[21] Chabries, D., Anderson, D., Jr., T. S., and Christiansen, R., "Application of a human auditory model to loudness perception and hearing compensation," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1995.

[22] Chakrabartty, S. and Cauwenberghs, G., "Fixed-Current method for Programming Large Floating-gate Arrays," in *Proceedings of the International Symposium on Circuits and Systems*, pp. 3934–37, May 2005.

[23] Chawla, R., Adil, F., Serrano, G., and Hasler, P., "Programmable Gm-C Filters using Floating-gate Operational Transconductance Amplifiers," *IEEE Transactions on Circuits and Systems I*, vol. 54, pp. 481–91, Mar 2007.

[24] Chawla, R., Bandyopadhyay, A., Srinivasan, V., and Hasler, P., "A 531 nW/MHz, 128X32 current-mode programmable analog vector-matrix multiplier with over two decades of linearity," in *Proceedings of the IEEE Custom Integrated Circuits Conference*, pp. 651–4, Oct 2004.

[25] Chen, R., Hung, T., and Hung, C., "A CMOS Infrared Wireless Optical Receiver Front-end with a variable gain fully-differential transimpedance amplifier," *IEEE Transactions on Consumer Electronics*, vol. 51, pp. 424–429, May 2005.

[26] Chicca, E., Badoni, D., Dante, V., D'Andreagiovanni, M., Salina, G., Carota, L., Fusi, S., and Giudice, P. D., "A VLSI Recurrent Network of Integrate-and-Fire Neurons Connected by Plastic Synapses With Long-Term Memory," *IEEE Transactions on Neural Networks*, vol. 14, pp. 1297–1307, September 2003.

[27] Chow, S.-N. and Zhou, H.-M., "An analysis of phase noise and Fokker-Planck equations," *Journal of Differential Equations*, vol. 234, pp. 391–411, Mar 2007.

[28] Ciofi, C., Crupi, F., Pace, C., and Scandurra, G., "Improved trade-off between noise and bandwidth in op-amp based transimpedance amplifier," in *IEEE Instrumentation and Measurement Technology Conference*, pp. 1990–3, May 2004.

[29] Cohen, M. and Cauwenberghs, G., "Floating-gate adaptation for focal-plane online nonuniformity correction," *IEEE Transactions on Circuits and Systems II*, vol. 48, pp. 83–89, Jan. 2001.

[30] Cowan, G., Melville, R., and Tsividis, Y., "A vlsi analog computer/digital computer accelerator," *IEEE Journal of Solid-State Circuits*, vol. 41, pp. 42–53, Jan 2006.

[31] Crawford, J., "Introduction to bifurcation theory," *Review of Modern Physics*, vol. 63, pp. 991–1037, 1991.

[32] de Langen, K.-J. and Huijsing, J. H., "Compact Low-Voltage Power-Efficient Operational Amplifier Cells for VLSI," *IEEE Journal of Solid-State Circuits*, vol. 33, pp. 1482–96, Oct 1998.

[33] Delbruck, T. and Mead, C., "Adaptive photoreceptor with wide dynamic range," in *Proceedings of the International Symposium on Circuits and Systems*, vol. 4, pp. 339–3542, June 1994.

[34] Delbruck, T. and Oberhoff, D., "Self-biasing low power adaptive photoreceptor," in *Proceedings of the International Symposium on Circuits and Systems*, vol. 4, pp. 844–7, May 2004.

[35] Doedel, E. J., "AUTO: A program for the automatic bifurcation and analysis of autonomous systems," *Congressus Numerantium*, vol. 30, pp. 265–84, 1981.

[36] Dudek, P. and Hicks, P., "A General-Purpose Processor-per-Pixel Analog SIMD Vision Chip," *IEEE Transactions on Circuits and Systems I*, vol. 52, pp. 13–20, Jan 2005.

[37] Farquhar, E. and Hasler, P., "A Bio-Physically Inspired Silicon Neuron," *IEEE Transactions on Circuits and Systems I*, vol. 52, pp. 477–488, March 2005.

[38] Folowosele, F. and et. al., A. H., "A switched capacitor implementation of the generalized linear integrate-and-fire neuron," in *Proceedings of the International Symposium on Circuits and Systems*, pp. 2149–52, May 2009.

[39] Galán, R. C., Jiménez-Garrido, F., Dominguez-Castro, R., Espejo, S., Roska, T., Rekeczsky, C., Petrás, I., and Rodriguez-Vázquez, A., "A Bio-Inspired Two-Layer Mixed-Signal Flexible Programmable Chip for Early Vision," *IEEE Transactions on Neural Networks*, vol. 14, pp. 1313–36, Sept 2003.

[40] Georgiou, J., Drakakis, E., Toumazou, C., and Premanoj, P., "An analogue micropower log-domain silicon circuit for the Hodgkin and Huxley nerve axon," in *Proceedings of the International Symposium on Circuits and Systems*, pp. 286–289, May 1999.

[41] Gordon, C., Farquhar, E., and Hasler, P., "A family of floating-gate adapting synapses based upon transistor channel models," in *Proceedings of the International Symposium on Circuits and Systems*, pp. 23–26, May 2004.

[42] Graham, D., Farquhar, E., Degnan, B., Gordon, C., and Hasler, P., "Indirect Programming of Floating-Gate Transistors," *IEEE Transactions on Circuits and Systems I*, vol. 54, pp. 951–63, May 2007.

[43] Graham, D., Hasler, P., Chawla, R., and Smith, P., "A Low-Power Programmable Bandpass Filter Section for Higher Order Filter Applications," *IEEE Transactions on Circuits and Systems I*, vol. 54, pp. 1165–76, June 2007.

[44] Guckenheimer, J. and Holmes, P., *Nonlinear Oscillations, Dynamical Systems and Bifurcations of Vector Fields*. New York: Springer-Verlag, Applied mathematical Sciences 42, 1983.

[45] Hasler, P., Koziol, S., Farquhar, E., and Basu, A., "Transistor Channel Dendrites implementing HMM classifiers," in *Proceedings of the International Symposium on Circuits and Systems*, pp. 3359–62, May 2007.

[46] Hayes, M., "A Nonlinear Optical Preamplifier for Sensing Applications," *IEEE Transactions on Circuits and Systems I*, vol. 49, pp. 1–9, Jan 2002.

[47] Hodgkin, A. and Huxley, A., "A Quantitative Description of Membrane Current and its Application to Conduction and Escitation in Nerve," *Journal of Physiology*, vol. 117, pp. 500–544, 1952.

[48] Hodgkin, A. L., "The local electric changes associated with repetitive action in a non-medulated axon," *Journal of Physiology*, vol. 107, pp. 165–181, 1948.

[49] Hoe, D. and Ribner, D., " An Auto-ranging photodiode preamplifier with 114dB Dynamic Range," *IEEE Journal of Solid-State Circuits*, vol. 31, pp. 187–194, Feb 1996.

[50] Hyde, J., Humes, T., Diorio, C., Thomas, M., and Figueroa, M., " A 300-MS/s 14-bit digital-to-analog converter in logic CMOS," *IEEE Journal of Solid-State Circuits*, vol. 38, pp. 734–40, May 2003.

[51] Indiveri, G., Chicca, E., and Douglas, R., "A VLSI Array of Low-Power Spiking Neurons and Bistable Synapses With Spike-Timing Dependent Plasticity," *IEEE Transactions on Neural Networks*, vol. 17, pp. 211–221, Jan 2006.

[52] Izhikevich, E., "Weakly Pulse-Coupled Oscillators, FM Interactions, Synchronization, and Oscillatory Associative Memory," *IEEE Transactions on Neural Networks*, vol. 10, pp. 508–526, May 1999.

[53] Izhikevich, E., "Neural excitability, spiking and bursting," *International Journal of Bifurcations and Chaos*, vol. 10, pp. 1171–1266, 2000.

[54] Izhikevich, E., "Which Model to Use for Cortical Spiking Neurons?," *IEEE Transactions on Neural Networks*, vol. 15, pp. 1063–1070, September 2004.

[55] Izhikevich, E. M., *Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting*. Cambridge, MA: MIT Press, 2007.

[56] Jung, R., Brauer, E., and Abbas, J., "Real-time interaction between a neuromorphic electronic circuit and the spinal cord," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 9, pp. 319–326, Sept 2001.

[57] Khorramabadi, H., Tzeng, L., and Tarsia, M., "A 1.06 Gb/s-31 dBm to 0 dBm BiCMOS optical preamplifier featuring adaptive transimpedance," in *Proceedings of the International Solid State Circuits Conference*, pp. 54–55, Feb. 1995.

[58] Kinoshita, S., Morie, T., Nagata, M., and Iwata, A., "A PWM analog memory programming circuit for floating-gate MOSFETs with 75-μs programming time and 11 bit updating resolution," *IEEE Journal of Solid-State Circuits*, vol. 36, pp. 1286–90, Aug 2001.

[59] Koch, C., *Biophysics of Computation: Information Processing in Single Neurons*. Oxford University Press, USA, 2004.

[60] Kohno, T. and Aihara, K., "A MOSFET-based model of a class 2 nerve membrane," *IEEE Transactions on Neural Networks*, vol. 16, pp. 754–773, May 2005.

[61] Kramer, J., "An integrated optical transient sensor," *IEEE Transactions on Circuits and Systems II*, vol. 49, pp. 612–628, Sept 2002.

[62] Laiho, M., Paasio, A., Kananen, A., and Halonen, K., "A Mixed-Mode Polynomial Cellular Array Processor Hardware Realization," *IEEE Transactions on Circuits and Systems I*, vol. 51, pp. 286–297, Feb 2004.

[63] Lecar, H., "Morris-Lecar model," *Scholarpedia*, vol. 2, no. 10, p. 1333, 2007.

[64] Lee, C. Y., "An algorithm for path connection and its application," *IRE Trans. Electronic Computer*, vol. 10, pp. 346–65, 1961.

[65] Lee, E. and Gulak, P., "A CMOS Field-Programmable Analog Array," *IEEE Journal of Solid-State Circuits*, vol. 26, pp. 1860–67, Dec 1991.

[66] Lee, E. K. and Gulak, P., "Field Programmable Analogue Array based on mosfet transconductors," *Electronics Letters*, vol. 28, pp. 28–29, Jan 1992.

[67] Livi, P. and Indiveri, G., "A current-mode conductance-based silicon neuron for address-event neuromorphic systems," in *Proceedings of the International Symposium on Circuits and Systems*, pp. 2898–2901, May 2009.

[68] Looby, C. A. and Lyden, C., "Op-amp based CMOS field-programmable analogue array," *IEE Proc.-Circuits Devices Systems*, vol. 47, pp. 93–99, Apr 2000.

[69] Mahowald, M. and Douglas, R., " A Silicon Neuron," *Nature*, vol. 354, pp. 515–518, Dec 1991.

[70] Masson, S. L., Laflaquiere, A., Bal, T., and Masson, G. L., "Analog Circuits for Modeling Biological Neural Networks: Design and Applications," *IEEE Transactions on Biomedical Engineering*, vol. 46, pp. 638–645, June 1999.

[71] Meyer, R. and Mack, W., " A Wideband Low-Noise Variable-Gain BiCMOS Transimpedance Amplifier," *IEEE Journal of Solid-State Circuits*, vol. 29, pp. 701–706, June 1994.

[72] Minch, B., "A low-voltage MOS cascode bias circuit for all current levels," in *Proceedings of the International Symposium on Circuits and Systems*, vol. 3, pp. 619–22, May 2002.

[73] Minch, B., "Synthesis of static and dynamic multiple-input translinear element networks," *IEEE Transactions on Circuits and Systems I*, vol. 51, pp. 409–421, Feb 2004.

[74] M.N.Ericson, K.G.Falter, and Rochelle, J., " A Wide-Range Logarithmic Electrometer with Improved Accuracy and Temperature Stability," *IEEE Trans. on Instrumentation and Measurement*, vol. 54, pp. 968–73, Dec 1992.

[75] nán Cembrano, G. L., Rodriguez-Vázquez, A., Galán, R. C., Jiménez-Garrido, F., Espejo, S., and Dominguez-Castro, R., "A 1000 FPS at 128X128 Vision Processor With 8-Bit Digitized I/O," *IEEE Journal of Solid-State Circuits*, vol. 39, pp. 1044–55, July 2004.

[76] Pankiewicz, B., Wojcikowski, M., Szczepanski, S., and Sun, Y., "A Field Programmable Analog Array for CMOS Continuous-Time OTA-C Filter Applications," *IEEE Journal of Solid-State Circuits*, vol. 37, pp. 125–136, Feb 2002.

[77] Patel, G., Cymbalyuk, G., Calabrese, R., and DeWeerth, S., "Bifurcation Analysis of a Silicon Neuron," in *Proceedings of the Neural Information Processing Systems*, pp. 731–737, 1999.

[78] Peng, S., Gurun, G., Twigg, C., Qureshi, M., Basu, A., Brink, S., Hasler, P., and Degertekin, F., "A Large-Scale Reconfigurable Smart Sensory Chip," in *Proceedings of the International Symposium on Circuits and Systems*, 2009.

[79] Peng, S., Qureshi, M., Basu, A., Guldiken, R., Degertekin, L., and Hasler, P., "Floating-Gate Based CMUT Sensing Circuit Using Capacitive Feedback Charge Amplifier," in *Proceedings of the Ultrasonic Symposium*, pp. 2425–28, Oct. 2006.

[80] Peng, S., Qureshi, M., Basu, A., Hasler, P., and Degertekin, L., "A Floating-gate Based Low-Power Capacitive Sensing Interface Circuit," in *Proceedings of the IEEE Custom Integrated Circuits Conference*, pp. 257–60, Sept 2006.

[81] Peng, S., Qureshi, M., Hasler, P., Basu, A., and Degertekin, L., "A Charge-Based Low-Power High-SNR Capacitive Sensing Interface Circuit," *IEEE Transactions on Circuits and Systems I*, vol. 55, pp. 1863–72, Aug 2008.

[82] Petre, C., Schlottman, C., and Hasler, P., "Automated conversion of Simulink designs to analog hardware on an FPAA," in *Proceedings of the International Symposium on Circuits and Systems*, pp. 500–503, May 2008.

[83] Phang, K. and Johns, D., " A CMOS Optical Preamplifier for Wireless Infrared Comunications," *IEEE Transactions on Circuits and Systems II*, vol. 46, pp. 852–859, July 1999.

[84] Preyer, A. and Butera, R., "Causes of Transient Instabilities in the Dynamic Clamp," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 17, pp. 190–98, April 2009.

[85] Quan, X., Embabi, S., and Sanchez-Sinencio, E., "A Current-Mode Based Field Programmable Analog Array Architecture for Signal Processing Applications," in *Proceedings of the IEEE Custom Integrated Circuits Conference*, pp. 277–80, May 1998.

[86] Ravindran, S., *Physiologically motivated methods for audio classification*. PhD thesis, Georgia Institute of Technology, Atlanta, GA, 2006.

[87] Rinzel, J. and Ermentrout, G., *Analysis of Neural excitability and oscillations*. Cambridge, MA: Methods in Neural Engineering, C. Koch and I. Segev, Eds, MIT Press, 1989.

[88] Ruotsalainen, T., Palojarvi, P., and Kostamovaara, J., "A current-mode gain-control scheme with constant bandwidth and propagation delay for a transimpedance preamplifier," *IEEE Journal of Solid-State Circuits*, vol. 34, pp. 253–258, Feb 1999.

[89] Saighi, S., Tomas, J., Bornat, Y., and Renaud, S., "A Conductance-Based Silicon Neuron with Dynamically Tunable Model Parameters," in *Proceedings of the International IEEE EMBS Conference on Neural Engineering*, pp. 285–88, 2005.

[90] Sarpeshkar, R., T.Delbruck, and Mead, C., " White noise in MOS transistors and resistors," *IEEE Transactions on Electron Devices*, vol. 9, pp. 23–29, Nov 1993.

[91] Sarpeshkar, R., *Efficient precise computation with noisy components: extrapolating from an electronic cochlea to the brain*. PhD thesis, California Institute of Technology, Pasadena, CA, 1997.

[92] Serrano, G. and Hasler, P., "A Precision Low-TC Wide-Range CMOS Current Reference," *IEEE Journal of Solid-State Circuits*, vol. 43, pp. 558–565, Feb. 2008.

[93] 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," in *Proceedings of the International Symposium on Circuits and Systems*, vol. 1, pp. 373–376, May 2004.

[94] Shevchuk, E. and Choi, K., "Folded cascode CMOS mixer design and optimization in 70 nm technology," pp. 943–946, Aug. 2005.

[95] Simoni, M., Cymbalyuk, G., Sorensen, M., Calabrese, R., and DeWeerth, S., "A Multiconductance Silicon Neuron With Biologically Matched

Dynamics," *IEEE Transactions on Biomedical Engineering*, vol. 51, pp. 342–354, Feb 2004.

[96] Simoni, M. and DeWeerth, S., "Adaptation in a VLSI Model of a Neuron," *IEEE Transactions on Circuits and Systems II*, vol. 46, pp. 967–970, July 1999.

[97] Skinner, F., Kopell, N., and Marder, E., "Mechanisms for oscillation and frequency control in networks of mutually inhibitory relaxation oscillators," *Journal of Computational Neuroscience*, vol. 1, pp. 69–87, 1994.

[98] S.P.DeWeerth, "Analog vlsi circuits for stimulus localization and centroid computation," *International Journal of Computer Vision*, vol. 8, pp. 191–202, Sept 1992.

[99] Srinivasan, V., *Programmable Analog Techniques for Precision Analog Circuits, Low-Power Signal Processing and On-Chip Learning*. PhD thesis, Georgia Institute of Technology, Atlanta, GA, Aug. 2006.

[100] Srinivasan, V., Serrano, G., Gray, J., and Hasler, P., "A Precision CMOS Amplifier Using Floating-Gate Transistors for Offset Cancellation," *IEEE Journal of Solid-State Circuits*, vol. 42, pp. 280–91, Feb. 2007.

[101] Srinivasan, V., Serrano, G., Twigg, C., and Hasler, P., "A Floating-gate based programmable CMOS reference," *IEEE Transactions on Circuits and Systems I*, vol. 55, pp. 3448–56, Dec. 2008.

[102] Takemoto, T., Kohno, T., and Aihara, K., "MOSFET implementation of class I* neurons coupled by gap junctions," *Artificial Life and Robotics*, vol. 10, no. 1, pp. 1–5, 2006.

[103] Takemoto, T., Kohno, T., and Aihara, K., "Circuit Implementation and Dynamics of a two-dimensional MOSFET Neuron Model," *International Journal of Bifurcations and Chaos*, vol. 17, pp. 459–508, 2 2007.

[104] Toygur, L., Yu, X., and Garverick, S., "High temperature, low-power 8-MegΩ 1.2 MHz SOI-CMOS transimpedance amplifier for MEMS-based wireless sensors," in *IEEE international SOI Conference*, pp. 179–81, Oct. 2004.

[105] Tran, H., Blyth, T., and et. al, D. S., "A 2.5V 256-level non-volatile analog storage device using EEPROM technology," in *Proceedings of the International Solid State Circuits Conference*, pp. 270–1, Feb. 1996.

[106] Tsividis, Y., *Operation and Modeling of the MOS Transistor*. Boston, MA: McGraw-Hill, 2002.

[107] Tsividis, Y., Satyanarayana, S., and Graf, H. P., "A reconfigurable analog VLSI neural network chip," in *Proceedings of the Neural Information Processing Systems*, pp. 758–768, 1990.

[108] Twigg, C. and Hasler, P., "A Large-Scale Reconfigurable Analog Signal Processor (RASP) IC," in *Proceedings of the IEEE Custom Integrated Circuits Conference*, pp. 5–8, Sept 2006.

[109] Twigg, C. and Hasler, P., "Programmable Conductance Switches for FPAAs," in *Proceedings of the International Symposium on Circuits and Systems*, pp. 173–76, May 2007.

[110] van den Bosch, A., Borremans, M., Steyaert, M., and Sansen, W., "A 10-bit 1-GSample/s Nyquist current-steering CMOS D/A converter," *IEEE Journal of Solid-State Circuits*, vol. 36, pp. 315–24, Mar 2001.

[111] van den Broeke, L. and Nieuwkerk, A., " Wide-band integrated optical receiver with improved dynamic range using a current switch at the input," *IEEE Journal of Solid-State Circuits*, vol. 28, pp. 862–64, July 1993.

[112] Vogelstein, R., Mallik, U., Vogelstein, J., and Cauwenberghs, G., "Dynamically Reconfigurable Silicon Array of Spiking Neurons With Conductance-Based Synapses," *IEEE Transactions on Neural Networks*, vol. 18, pp. 253–265, Jan 2007.

[113] Vogelstein, R., Tenore, F., Guevremont, L., Etienne-Cummings, R., and Mushahwar, V. K., "A Silicon Central Pattern Generator controls Locomotion in vivo," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 2, pp. 212–222, Sept 2008.

[114] Wong, Y., Cohen, M., and Abshire, P., "A Floating-gate Comparator with Automatic Offset Adaptation for 10-bit Data Conversion," *IEEE Transactions on Circuits and Systems I*, vol. 52, pp. 1316–26, July 2005.

[115] Yu, T. and Cauwenberghs, G., "Analog VLSI neuromorphic network with programmable membrane channel kinetics," in *Proceedings of the International Symposium on Circuits and Systems*, pp. 349–52, May 2009.

VITA

Arindam Basu received the B.Tech and M.Tech degrees in Electronics and Electrical Communication Engineering from the Indian Institute of Technology, Kharagpur, India in 2004 and 2005 respectively. He received the M.S. degree in mathematics from the Georgia Institute of Technology in 2009 and the PhD. degree in Electrical Engineering in 2010 from the same school. His research interests include non-linear dynamics and chaos, modeling neural dynamics, low power analog IC design and programmable circuits and devices. Mr. Basu received the JBNSTS award in 2000 and the Prime Minister of India Gold Medal in 2005 from I.I.T Kharagpur. He is also the co-recipient of the best student paper award in the Ultrasonics symposium in 2006 and a finalist in the best student paper contest at ISCAS 2008.