Floating-Gates Transistors For Precision Analog Circuit Design: An Overview

Venkatesh Srinivasan*, David W. Graham†and Paul Hasler‡

School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, Georgia 30332–0250
Email: *vsriniva@ece.gatech.edu; †dgraham@ece.gatech.edu; ‡phasler@ece.gatech.edu

***Abstract*— This paper presents an overview of floating-gate transistors with an emphasis on using them as programmable elements to correct mismatch inherent in analog circuit design. The design methodology is such that floating-gate MOSFETs play the role of programmable elements while forming an inherent part of the circuitry of interest, as well. Such an approach results in a compact architecture with minimal additional power. Accurate programming that is key to a successful implementation is discussed along with experimental results demonstrating floating-gate charge retention. An overview of several circuit design examples is provided to demonstrate the feasibility of using floating-gate transistors for precision analog circuit design.**

I. Mismatch in Analog Circuits

Precision analog circuit design has been limited primarily due to mismatch in integrated circuit components such as transistors, resistors and capacitors. Offsets, by way of mismatch within amplifiers, limits the available input signal dynamic range. Offsets in comparators place a lower limit on available signal resolution. Matching between transistors directly impact the achievable accuracy in current-mode digital-to-analog converters. Also, mismatch is the key issue when designing high-accuracy analog-to-digital converters and precision references.

Component mismatch has been addressed through the use of larger devices and layout techniques such as common-centroid layout. From a design perspective, offsets in amplifiers have been addressed using schemes such as autozeroing, correlated double sampling and chopper stabilization [1]. In analog-to-digital converters, it is common to use digital calibration to correct for errors due to mismatch [2]. Continuous-time filters employ elaborate tuning schemes to account for variations in transconductance and capacitance [3]. Other schemes include laser trimming and use of poly fuses. In all of the above techniques, mismatch is corrected at the cost of area, power and design complexity. In this paper, the issue of mismatch in analog circuit design is addressed from the context of using floating-gate transistors as programmable elements with minimal additional overhead.

The design methodology uses floating-gate transistors as programmable elements that correct mismatch in analog circuitry. Rather than using floating-gate MOSFETs as seperate trimming elements, transistors that are an inherent part of the circuit architecture are designed to be floating gates such that circuit imperfections due to mismatch can be accounted for through programming. Such an approach reduces design overhead and results in a compact architecture with minimal extra power dissipation. Also, the non-volatile charge retention of floating gates obviates the need for constant refresh cycles.

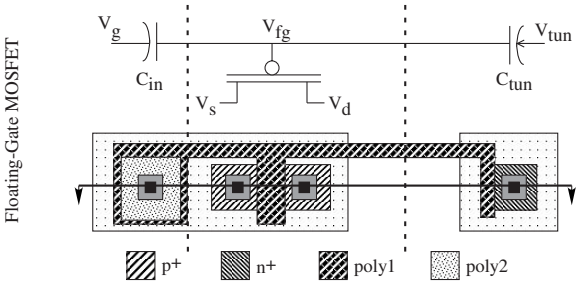

Fig. 1. **Floating-Gate pFET**: Layout and circuit schematic of a floating-gate pFET in a standard double poly n-well CMOS process. External input signals are capacitively coupled onto the floating gate through the input capacitor C_{in}. The layout shows the realization of C_{in} using a poly-poly capacitor, although this could also be implemented as a MOS capacitor or a diffused linear capacitor. Between V_{tun} and the floating gate is the tunneling capacitor C_{tun}.

Floating-gate transistors are introduced in section II along with schemes for modifying the charge on the floating gate. Programming accuracy and the achievable precision are discussed, as well as charge retention in floating gates. Section III demonstrates the design approach through the use of floating gates for offset cancellation in differential pairs that form the basis for operational amplifiers. An overview of floating-gate transistors finding applications in other areas such as D/A converters, flash A/D converters, $\Sigma-\Delta$ converters and continuous-time filters is presented, as well. Finally, conclusions, along with future directions, are presented in section IV.

II. Floating-Gate MOS Transistor

A floating-gate MOS transistor is similar to a standard MOS device except that its gate is completely surrounded by SiO_2, a high quality insulator. This creates a potential barrier that prevents charge stored on the floating gate from leaking off of the floating node. In circuit terms, a floating gate is one that has no DC path to ground. Figure 1 shows the circuit schematic and layout of a floating-gate pMOS transistor. To ensure non-volatile storage of charge on the floating gate, external inputs are capacitively coupled through an input capacitor C_{in}. This capacitor is realized as a poly-poly capacitor using the second polysilicon layer shown in Fig. 1. The additional capacitance shown, namely, C_{tun} is used for Fowler-Nordheim tunneling that modifies the charge on the floating gate.

A. Programming Floating-Gate Transistors

Programming a floating-gate transistor involves setting the DC voltage of the floating node to any desired value by adding to or removing charge from the floating gate. Charge modification is achieved by exploiting two physical phenomena, namely hot-electron injection and Fowler-Nordheim tunneling [4]. Tunneling-based programming involves the application of high-voltage pulses of both positive and negative polarities to modify floating-gate charge. The logarithmic nature of tunneling, however, makes faster programming highly time consuming [5]. Faster programming times can be achieved by using special processing steps such as an ultra-thin tunneling oxide or textured polysilicon or achieved by increasing the tunneling voltage even further.

Hot-electron injection,on the other hand, is a phenomenon that involves adding electrons onto the floating gate. Hot-electron injection occurs when the electric field in the channel is high enough that it accelerates channel electrons to energies higher than the $Si - SiO_2$ barrier. In a pFET where the carriers are holes, electrons are produced by a hot-hole impact ionization. A detailed analysis of the physics of hot-electron injection in pFETs is presented in [6]. Qualitatively, the number of electrons injected onto the gate of a pFET depends on the source-to-drain voltage, the drain current and the time for which the source-to-drain voltage is maintained higher than the value necessary for injection.

Fast and accurate programming is achieved by using a combination of hot-electron injection and tunneling, where tunneling is used primarily as a global erase for all floating gates in the circuit. Once the charge on all floating gates have been normalized, hot-electron injection is used to individually program each floating gate to the desired value. This is achieved by first isolating the floating-gate transistor from the rest of the circuitry and then applying a sufficient source-to-drain voltage for a specific period of time that is based on the desired floating-gate target current. A detailed discussion of the programming scheme is given in [7], [8]. Figure 2 shows the programming of a pFET floating-gate device in a $0.5\mu m$ standard digital CMOS process with the power supply set at $3.3V$. As can be observed, the threshold voltage of the device can be programmed from $0.75V - 2.75V$ using the programming methodology described above. Injection decreases the threshold voltage of a pFET while tunneling increases the threshold voltage as shown.

B. Programming precision

Figure 3 shows the programming accuracy that is currently achievable [8]. The accuracy to which one can program floating-gate transistors to a target current depends on the smallest drain current change that can be programmed onto a floating-gate device. In order to estimate the design choices available to improve programming precision, let us assume that the floating-gate pFET device is operating in the sub-threshold regime. The drain current is therefore given by,

$$I = I_o exp\left(\frac{-\kappa V_g}{U_T}\right) exp\left(\frac{V_s}{U_T}\right) \tag{1}$$

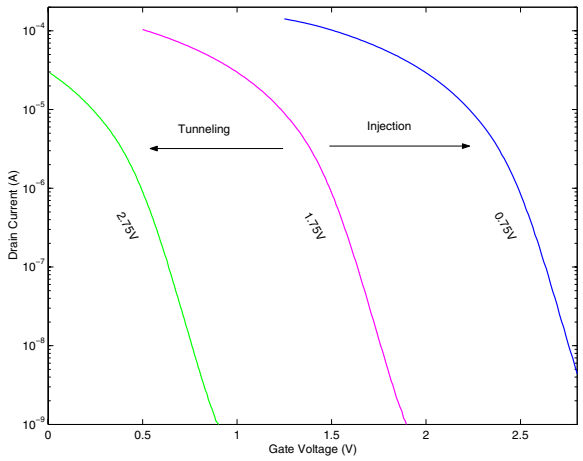

Fig. 2. **Floating-Gate pFET Programming**: Programming a floating-gate transistor using a combination of tunneling and hot-electron injection, the threshold voltage of a pFET is programmed to three different threshold voltages of $0.75V$, $1.75V$ and $2.75V$. The threshold voltage of a non-floating-gate device in the process is approximately $0.9V$.

where all the variables have their usual meaning. For a ΔV_g change in the gate voltage, a ΔI change in drain current results and the net programmed drain current of the device is given by,

$$I + \Delta I = I_o exp\left(\frac{-\kappa(V_g + \Delta V_g)}{U_T}\right) exp\left(\frac{V_s}{U_T}\right) \tag{2}$$

Dividing (2) by (1) gives,

$$\frac{\Delta I}{I} = exp\left(\frac{-\kappa \Delta V_g}{U_T}\right) - 1 \tag{3}$$

The change in floating-gate voltage is related to the programmed floating-gate charge as,

$$\Delta V = \frac{\Delta Q}{C_T} \tag{4}$$

where, C_T is the total capacitance at the floating-gate node and ΔQ is the programmed charge. Using (4) in (3), the achievable change in drain current due to programming relative to the initial drain current is given by,

$$\frac{\Delta I}{I} = exp\left(\frac{-\kappa \Delta Q}{U_T C_T}\right) - 1 \tag{5}$$

It is clear from (5) that the achievable precision is directly proportional to the charge that can be reliably transferred onto the floating gate and inversely proportional to the total floating-gate capacitance. Given that the theoretical minimum for charge transfer is equal to that of a single electron, in quantitative terms, if one assumes a floating-gate capacitance of $16fF$ (a small device), a κ of 0.7 and $U_T = 25mV$, a single electron change results in an accuracy of $2.8{\times}10^{-4}$ (12 bits) over the entire sub-threshold range of $6-8$ decades. If, however, the capacitance is increased by a factor of 10, the accuracy improves to $2.8{\times}10^{-5}$ or 15 bits.

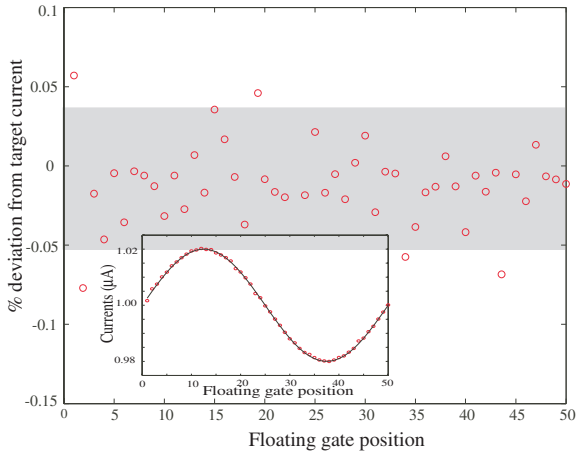

Fig. 3. **Programming precision**: Programming a $20nA$ sinusoid riding on a DC value of $1\mu A$ is shown along with the percentage error between the programmed current and the desired target. As can be observed, an error of $\pm 0.05\%$ has been achieved.

C. Floating-Gate Charge Retention

Charge loss in floating-gate transistors falls under two categories that occur due to different physical processes. These include a short-term drift that is observed immediately after programming and a long-term charge loss that occurs over years. The short-term drift in floating-gate charge has been attributed to the interface trap sites settling to a new equilibrium. Also, it has been observed that the drift is proportional to the amount of change that is programmed onto the floating gate. For instance, using the threshold voltage of the device as an indicator of the programmed change, the short-term drift in threshold voltage is proportional to the difference between the programmed threshold voltage and its initial value [9].

Long-term charge loss in floating-gate transistors occur due to a phenomenon of thermionic emission [10], [11], [12]. The amount of charge lost is a function of both temperature and time. Data extrapolated from accelerated temperature tests on floating-gate transistors, where floating gates have been exposed to high temperatures ($> 125°C$) for prolonged periods of time, indicate floating-gate charge loss of $< 1\%$ over a period of 10 years [9] thereby demonstrating excellent charge retention.

III. Floating-Gate Applications

Over the years, floating-gate transistors have found widespread use as programmable elements in analog circuit designs. A number of researchers have used floating gates as separate trimming elements to correct for offset voltages in amplifiers [9], [12]. However, using floating gates as both trimming elements and as an integral part of the circuitry of interest results in a compact architecture with minimal extra power dissipation and design overhead. To illustrate the design methodology, consider the floating-gate based differential pair shown in Fig. 4 [13]. Since offsets result due to the difference in currents through the transistors $M1$ and $M2$, on account of mismatch, the use of floating gates allow for correcting

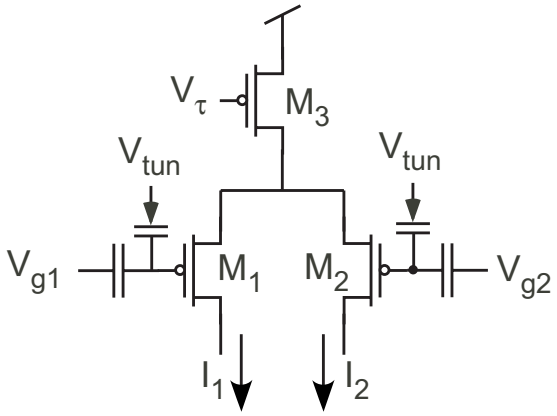

Fig. 4. **Floating-Gate Differential Pair**: Circuit schematic of a floating-gate based differential pair. The offsets resulting from current mismatch between transistors can be canceled by floating-gate programming.

the mismatch by programming the currents to be equal. It can be observed that floating gates play the dual role of both programming elements and a part of the circuitry of interest, thereby resulting in a compact architecture. The feasibility of the approach has been demonstrated in [13]. Such an offset cancellation scheme can be extended to operational amplifiers and comparators, as well. This has been demonstrated in [14] by way of a precision CMOS amplifier with floating-gate based offset cancellation.

The same principle of programming out offsets in differential pairs has been applied to a variety of other precision analog blocks. Data converters are a prime candidate for illustrating the uses of precise control of bias currents and voltages that can be achieved through floating-gate programming. A simple current-mode digital-to-analog converter (DAC) can be constructed from binary-weighted current sources with the digital bits turning the current sources either on or off. Typically, binary-weighted current sources are designed by scaling a unit-size transistor that is made large to counter the effects of device mismatch. Achieving high accuracy involves an area penalty. However, using floating-gate transistors for the current sources and programming the currents to the desired value results in an extremely compact architecture that enables the use of multiple converters on the same die [15].

Traditional design of charge amplifier DACs suffer from large element spread. Linearity is improved by minimizing the capacitor mismatch at the cost of using larger capacitors and, hence, increased area. A floating-gate based charge amplifier DAC that uses electronic potentiometers (e-pots) [16] in their implementation is described in [17]. Here, e-pots are used to set programmable on-chip reference voltages for the DAC such that capacitor mismatch can be accounted for through programming the e-pots. Such an approach allows for lower capacitance spread and accounts for capacitance mismatch, as well.

Floating gates find applications in analog-to-digital converters (ADC), as well. Flash ADCs are plagued by resistor mismatch and offsets in comparators that severely limit their achievable precision. However, by removing the ubiquitous

resistor chain that sets the reference voltages in the flash ADC and replacing it with a series of programmable reference voltages, offsets in comparators can be accounted for, thereby improving the accuracy of the overall converter. Such an approach has been demonstrated in [18] where e-pots have been used as programmable references. E-pots have also been used as part of the modulator loop in a $\Sigma - \Delta$ converter as tunable elements to set varying integrator gains. This provides the designer the flexibility to set, within limits, suitable noise shaping functions and thereby achieve higher signal-to-noise ratio [19], [20].

Continuous-time filters are another class of analog circuits that require tuning of their component parameters (typically g_m) to ensure satisfactory performance. Using floating-gate transistors as tuning elements in an OTA results in a programmable transconductance. These can then be used to construct continuous-time filters with tuning performed by programming floating-gate transistors. The approach has been demonstrated in [21] with the design of a floating-gate programmable OTA and its use in a second-order continuous-time g_mC filter.

IV. Conclusions

The use of floating-gate transistors in analog circuits gives the designer much-needed programmability, since floating-gate transistors provide non-volatile, but programmable, charge storage. With the semiconductor industry heading towards a 45nm node in 2010 [22], what does the future hold for programmable analog design using floating-gates? This question is prompted primarily due to valid concerns about charge leakage in thinner gate oxides ($< 5nm$). In answering, one needs to assess the general trends in mixed-signal designs. Current trends seem to suggest that future mixed signal systems are heading towards a multi-chip system-on-a-package solution. Such a trend would result in analog circuits designed in older, well-characterized processes while still being integrated with digital circuitry designed in the latest process technology. Also, most modern CMOS processes (starting with $0.35\mu m$ and $0.25\mu m$) provide thick-oxide devices that handle higher voltages ($3.3V, 5V$). These can now be used as floating-gate transistors on account of the thicker gate oxide. Additionally in cutting edge processes, floating-gate transistors with very thin oxides can utilize the gate current leakage for adaptive systems. Also, it is likely that new dielectric materials will be used in future processes that will allow all transistors, including floating-gate transistors, to operate with low leakage currents while still using the thin oxide. Therefore, one can conclude that floating-gate transistors will continue serving a useful role in future analog circuit designs.

V. Acknowledgements

This paper has been based heavily on the research conducted at the CADSP laboratory at Georgia Institute of Technology, Atlanta, GA. The authors would like to thank the members of the lab (both past and present) for valuable discussions that have made this paper possible.

References

[1] C. C. Enz and G. C. Temes, "Circuit techniques for reducing the effects of op-amp imperfections: autozeroing, correlated double sampling and chopper stabilization," *Proceedings of the IEEE*, vol. 84, pp. 1584–1614, Nov. 1996.

[2] J. Ming and S. H. Lewis, "An 8-bit 80-Msamples/s pipelined analog-to-digital converter with background calibration," *IEEE Journal of Solid-State Circuits*, vol. 36, pp. 1489–1497, Oct. 2001.

[3] F. Krummenacher and N. Joehl, "A 4-MHz continuous-time filter with on-chip automatic tuning," *IEEE Journal of Solid-State Circuits*, vol. 23, pp. 750–758, June 1998.

[4] M. Lezlinger and E. Snow, "Fowler-Nordheim tunneling in thermally grown SiO_2," *Journal of Applied Physics*, vol. 40, pp. 278–283, Jan. 1969.

[5] T. M. S. Kinoshita, M. Nagata, and A. Iwata, "A PWM analog memory programming circuit for floating-gate MOSFETs with $75\mu s$ programming time and 11-Bit updating resolution," *IEEE Journal of Solid-State Circuits*, vol. 36, pp. 1286–1290, May 2003.

[6] C. Duffy and P. Hasler, "Modeling hot-electron injection in PFET's," *Journal of Computational Electronics*, vol. 2, pp. 317–322, Dec. 2003.

[7] G. Serrano, P. Smith, H. Lo, R. Chawla, T. Hall, C. Twigg, and P. Hasler, "Automated rapid programming of large arrays of floating-gate elements," in *Proceedings of the International Symposium on Circuits and Systems*, vol. I, May 2004, pp. 373–376.

[8] A. Bandyopadyay, G. Serrano, and P. Hasler, "Programmaing analog computational elements to 0.2% accuracy over 3.5 decades using a predictive method," *Proceedings of the International Symposium on Circuits and Systems*, 2005.

[9] E. Sackinger and W. Guggenbuhl, "An analog trimming circuit based on a floating-gate device," *IEEE Journal of Solid-State Circuits*, vol. 23, pp. 1437–1440, Dec. 1988.

[10] H. Nozama and S. Kokyama, "A thermionic electron emission model for charge retention in SAMOS structures," *Japanese Journal of Applied Physics*, vol. 21, pp. L111–L112, Feb. 1992.

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

[12] L. R. Carley, "Trimming analog circuits using floating-gate analog MOS memory," *IEEE Journal of Solid-State Circuits*, vol. 24, pp. 1569–1574, Dec. 1989.

[13] F. Adil, G. Serrano, and P. Hasler, "Offset removal using floating gate circuits for mixed-signal systems," in *Southwest Symposium on Mixed-Signal Design*, Feb. 2003, pp. 190–195.

[14] V. Srinivasan, G. Serrano, J. Gray, and P. Hasler, "A precision CMOS amplifier using floating-gates for offset cancellation," *accepted to Proceeding of the IEEE Custom Integrated Circuits Conference*, 2005.

[15] G. Serrano and P. Hasler, "A floating-gate DAC array," *Proceedings of the International Symposium on Circuits and Systems*, vol. 1, pp. 357–360, May 2004.

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

[17] E. Ozalevli, P. Hasler, and F. Adil, "A programmable voltage-output floating-gate digital-analog converter," *Proceedings of the International Symposium on Circuits and Systems*, vol. 1, pp. 1064–1067, May 2004.

[18] P. Brady and P. Hasler, "Investigations using floating-gate circuits for flash ADCs," *Proceedings of the Mid-West Symposium on Circuits and Systems*, vol. 2, pp. 83–86, August 2002.

[19] A. Pereira, P. Brady, A. Bandyopadyay, and P. Hasler, "Experimental investigations of floating-gate circuits in $\Sigma-\Delta$ converters," *Proceedings of the Mid-West Symposium on Circuits and Systems*, pp. 208–211, Aug 2002.

[20] A. Pereira, D. Allen, and P. Hasler, "A $0.5\mu m$ CMOS programmable discrete-time $\Sigma-\Delta$ modulator with floating-gate elements," *Proceedings of the International Symposium on Circuits and Systems*, pp. I–213–I–216, May 2004.

[21] R. Chawla, G. Serrano, D. J. Allen, A. W. Pereira, and P. Hasler, "Fully differential floating-gate programmable OTAs with novel common-mode feedback," *Proceedings of the International Symposium on Circuits and Systems*, vol. 1, pp. 817–820, May 2004.

[22] "International technology roadmap for semiconductors - 2004," *http://public.itrs.net*.