

Below-Ground Injection of Floating-Gate Transistors for Programmable Analog Circuits

Mir Mohammad Navidi[†], David W. Graham[†], and Brandon Rumberg[‡]

[†] Lane Department of Computer Science and Electrical Engineering, West Virginia University

[‡] Aspinity, Inc., Morgantown, West Virginia 26505 USA

Email: david.graham@mail.wvu.edu

Abstract—Floating-gate transistors have been used in many systems to add non-volatile memory. By storing a precise amount of charge on the electrically isolated floating gate, these devices can be used in analog applications as programmable current sources. However, manipulating the stored charge must be done with voltage drops that exceed the supply voltage. In this paper, we present a method to program floating-gate transistors using negative voltages—a desirable characteristic that has been previously inhibited by the difficulties of operating selection circuitry at voltages below the substrate potential. We circumvent the need to have selection circuitry operating at negative voltages by using indirectly programmed floating-gate transistors and a circuit that linearizes the programming currents. We also present a charge pump to generate the necessary negative voltages for programming in this configuration.

I. INTRODUCTION

Because of their charge-retention properties, floating-gate (FG) transistors have found a wide range of applications including nonvolatile memory [1], programmable analog devices [2], ultra-low-power systems [3], and adaptive circuits [4]. FG transistors can be fabricated in standard CMOS processes by creating MOSFETs with no DC path to their gates. This “floating” gate can hold a specific charge for long periods of time, thereby giving these devices nonvolatile memory characteristics. For programmable analog applications, the stored charge must be set extremely precisely in order to accurately represent an analog value. The specific amount of charge on an FG can be modified using hot-electron injection and Fowler-Nordheim tunneling. The process of selecting an individual FG and precisely programming it requires significant infrastructural overhead, especially when considering that both tunneling and injection require on-demand generation of voltage drops greater than V_{DD} .

In this paper, we present a technique for simplifying the infrastructure for programming floating-gate transistors. Specifically, we present a technique to use “below-ground injection” that permits injection of individual FG transistors in a large array but does not require any isolation switches to operate below the substrate voltage. Consequently, this technique can be used in any standard single-well CMOS process to accurately program analog values on FG transistors. We also present a charge pump that can produce the negative voltages required for injection. The end result permits an overall system that requires far less infrastructure and power-consumption than a more-conventional system using voltages above V_{DD} to create injection conditions.

This material is based upon work supported by the National Science Foundation under Award No. 1148815.

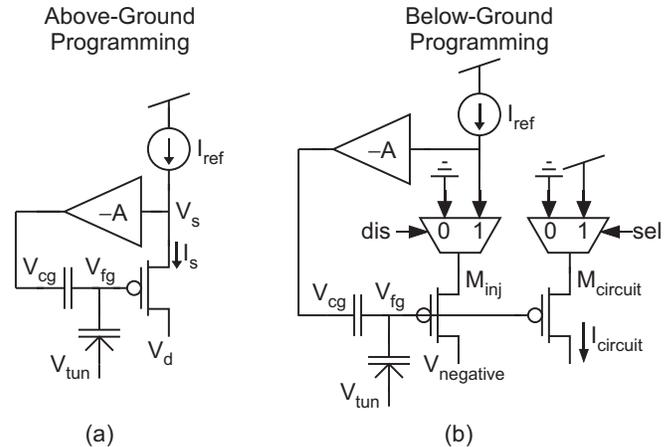


Fig. 1. (a) Continuous programming method using voltages above V_{DD} . (b) The proposed continuous programming method employing negative voltages.

All results were measured from an integrated circuit fabricated in a standard $0.5\mu\text{m}$ CMOS process.

II. BELOW GROUND PROGRAMMING

Two main techniques are used to program the charge on a floating-gate transistor. Fowler-Nordheim tunneling transports electrons across an oxide due to a large electric field. Because the required voltages for tunneling are large (typically greater than the junction breakdown voltage), high-voltage devices are required if trying to isolate/multiplex a specific FG for tunneling. Such high-voltage devices can be difficult to implement in standard CMOS processes. As a result, tunneling is reserved for global erasure in most programmable analog systems [5]; a single tunneling voltage is generated and applied to all FGs simultaneously to “erase” them.

Hot-electron injection can be used to add electrons to the floating gate. The voltages required for injection typically exceed the rated V_{DD} of the process but are generally lower than junction breakdown voltages. Consequently, standard transistors can be used to isolate/multiplex a specific FG for injection. Because of the ability to select a specific FG for injection, injection has become the preferred method for programming large arrays of FGs in analog applications [5].

PMOS devices have dominated analog floating-gate circuits since the injection currents necessary for programming can be created much more efficiently in PMOS devices than in NMOS devices. The amount of injection current in a PMOS device can

be represented by the simplified equation

$$I_{inj} \approx \beta I_s^\alpha e^{V_{sd}/V_{inj}} \quad (1)$$

where α , β , and V_{inj} are fit parameters. This equation covers the subthreshold operating range, which is where the injection operation is most efficient. Of note, the channel current (I_s) and the source-to-drain voltage (V_{sd}) determine the injection rate. As was mentioned earlier, significant injection only occurs for $V_{sd} > V_{DD}$.

When injecting a floating-gate transistor to a specific analog value, the channel current and V_{sd} must be precisely controlled. The technique of simply raising the source voltage to initiate injection is not ideal; this would alter V_{sg} , thereby changing the channel current to an unknown value. This simple technique makes it hard to achieve predictable and precise programming results.

Instead, the more conventional approach is to lower the drain potential. Because V_{sg} stays constant and because the drain has little influence on the channel current in saturation, the channel current remains at a known value. Lowering the drain potential below ground seems to be an obvious choice to generate the large V_{sd} required for injection; however, this has long been difficult to implement in most conventional systems. For example, when working with arrays of FG transistors, a multiplexer at the drain of the FG transistor is needed to connect the drain to a negative voltage in “program mode” but then connect the drain to its circuit during a “read mode” [5], [6]. In a single-well process, making selection circuitry operate below ground is not easily achieved, since the NMOS devices in a standard transmission gate would have their diffusion areas forward biased.

Instead of using negative voltages for injection, the common procedure for injecting FGs for analog applications is to raise the source and gate voltage up well above V_{DD} in a fashion that permits the channel current to be known (i.e. maintain a constant V_{sg}). Then, the drain can be at a positive voltage and still provide a V_{sd} that is large enough to invoke injection. This process of “ramping up” all voltages associated with the FG transistors and then lowering the drain to start injection requires significant infrastructure. Particularly if all of the programming infrastructure is to be included on-chip, the infrastructure should be made to be compact, low-power, and easy-to-use.

In the next Section, we present a simple circuit that permits below-ground injection, thereby greatly simplifying the necessary infrastructure. This circuit also eliminates the need for multiplexers at the drain of the FG transistor undergoing injection. In Section IV, we discuss a charge-pump circuit that can be used to generate the negative voltages on-chip, and in Section V, we show the results of this circuit and describe how it can be used in a large array.

III. FLOATING-GATE MEMORY CELL

In [6], we described a floating-gate memory cell that can be used to linearize the injection process, and a simplified version of this technique is illustrated in Fig. 1(a). An amplifier is used to modulate the control gate voltage, V_{cg} , of the FG transistor so that the floating node maintains a constant voltage during injection and/or tunneling—this negative feedback ensures that the source-to-floating-gate voltage remains constant so that

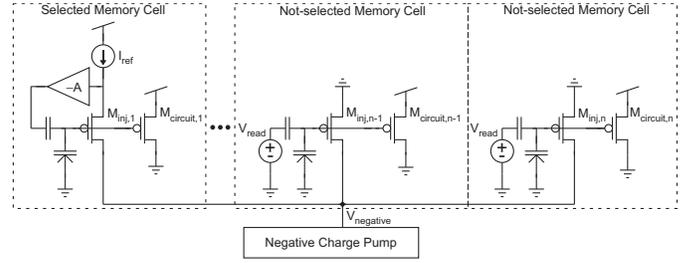


Fig. 2. Proposed technique to program a selected transistor in an array of FG transistors using a negative charge pump.

I_{ref} continually flows through the device. In [6], we used “above-ground” injection to program the FG transistor, by raising V_{DD} and then lowering the drain of the FG transistor toward ground. However, this technique still suffers from needing to first “ramp up” the FG transistor before programming, and the associated infrastructure to do so.

Figure 1(b) illustrates a simplified version of the memory cell proposed in this work. This new memory cell is an extension of our previous version [6] with the addition of a second transistor connected to the FG node. The current flowing through $M_{circuit}$ is directly set by the stored charge on the FG, and it is connected directly to a circuit to provide a bias current. All injection current goes through M_{inj} , so $M_{circuit}$ never needs to experience large V_{sd} values. Instead, the drain of M_{inj} is lowered below ground to induce injection. A multiplexer at the source of M_{inj} is used to disconnect M_{inj} from the feedback amplifier when the transistor is not actively being programmed. As a result, many FG transistors can all be connected to a single drain node that is lowered below ground, and injection can be prevented in a non-selected FG by connecting its source to ground. This technique permits the selection of an individual FG transistor for programming, and no selection switch needs to operate below ground.

Figure 2 shows a block diagram of an FG array when a specific memory cell is being programmed. In this method, the drain connection of all M_{inj} transistors in all memory cells will be connected to the output of a negative charge pump. The charge pump is designed such that the output voltage is either at V_{DD} when disabled or a negative value when enabled. When a specific memory cell requires injection, the multiplexer inside that specific memory cell will be configured such that the feedback loop is closed. At the same time, the sources of all the other M_{inj} transistors in the array will be connected to ground through the multiplexers. Therefore, when a negative voltage is applied to $V_{negative}$ only one transistor has a V_{sd} large enough to generate injection.

IV. NEGATIVE CHARGE PUMP

The negative voltage used in the programming phase is generated through a switched-capacitor charge pump. A block diagram of the regulated charge pump that generates the negative voltage for the programmer is shown in Fig. 3(a). This charge pump is based on our work in [7] which uses a variable-frequency regulation technique to provide a low-ripple output voltage. Our charge pump uses 4 charge transfer switch stages, which are shown in Fig. 3(b). A voltage divider shifts the output voltage up to positive values, and the difference between this voltage (V_{fb}) and the target voltage (V_{targ}) is

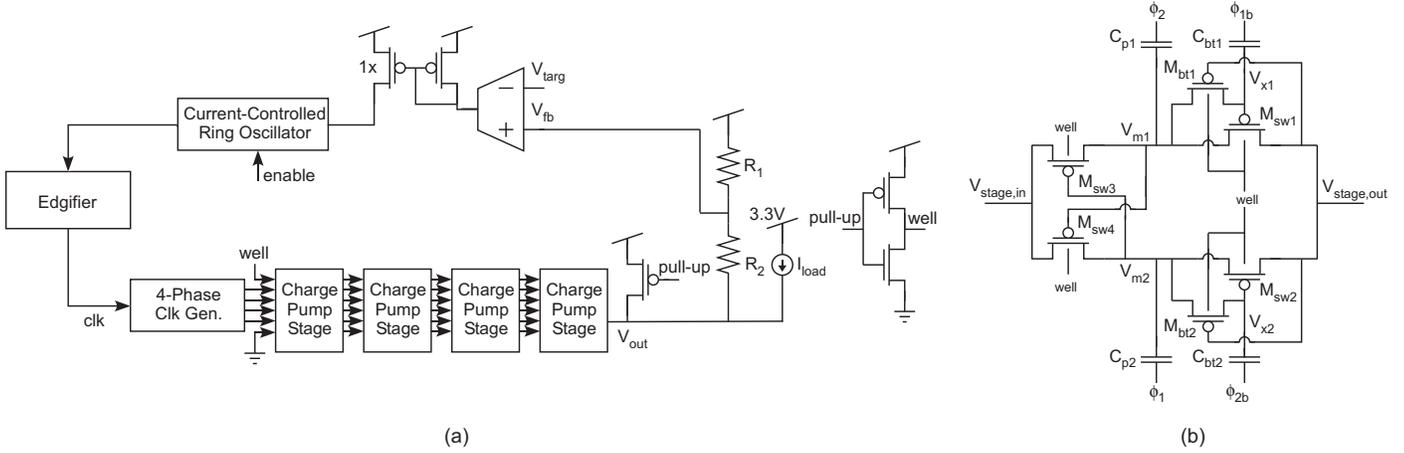


Fig. 3. (a) Block diagram of the negative charge pump. Under typical operation, enable and pull-up are tied together. (b) Circuit diagram of the charge pump stage from (a).

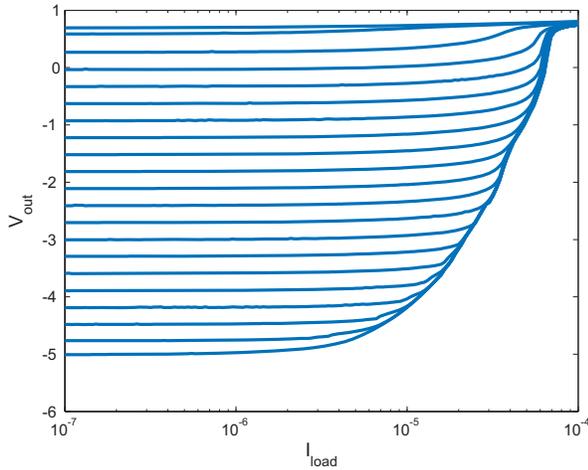


Fig. 4. Line regulation measurements of the negative charge pump.

converted to a current. This current modulates the frequency of the current-controlled ring oscillator. The relation between V_{targ} and V_{out} can be expressed as:

$$V_{out} = \left(1 + \frac{R_2}{R_1}\right) V_{targ} - \frac{R_2}{R_1} V_{DD} \quad (2)$$

In this particular implementation, we used $V_{DD} = 3.3V$, $R_1 = 300K\Omega$, and $R_2 = 600K\Omega$. Figure 4 shows the line regulation measurement results for output voltages from $-5V$ to $1V$ and a load-current range from $100nA$ to $100\mu A$. In this charge pump, the enable input of the oscillator and pull-up node are tied together. This will set the charge pump output to V_{DD} when the charge pump is disabled.

V. PROGRAMMING FLOATING-GATE TRANSISTORS USING NEGATIVE VOLTAGES

Figure 5 shows our complete programming circuit that permits using below-ground voltages for inducing injection. The left-hand side of Fig. 5 is a compact version of our circuit for linearizing injection (Fig. 1(b)); a common-source amplifier (M_1 - I_2) replaces the high-gain inverting amplifier in

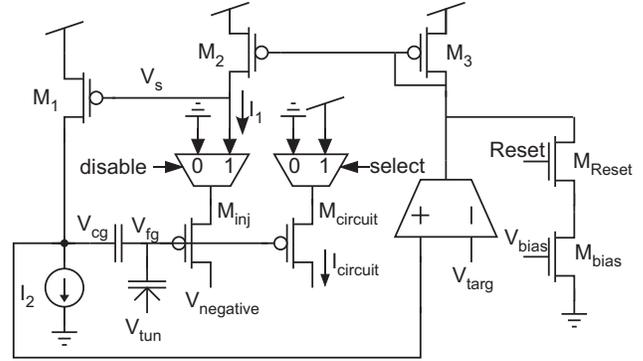


Fig. 5. Complete below-ground programming structure.

Fig. 1(b), and M_2 acts as the current source I_{ref} . Additionally, an operational transconductance amplifier (OTA) is connected in a feedback loop to stop injection when V_{cg} reaches a desired target voltage, V_{targ} . Transistors M_{Reset} and M_{bias} are used for initiating injection, as will be discussed shortly.

To program the FG to a desired target, the circuit is placed in “program mode,” which is done by setting “disable” to connect the source of M_{inj} to V_s and by setting “select” to connect the source of $M_{circuit}$ to ground. When the negative charge pump is enabled, there will be a large V_{sd} across M_{inj} causing injection currents, but the V_{sd} across $M_{circuit}$ will be too small to cause any injection. Assuming that the FG has been tunneled prior to injecting, node V_{cg} will ramp linearly up as electrons are injected onto the FG via injection through M_{inj} . While V_{cg} is well below the desired V_{targ} , the OTA will supply a constant current equal to its bias current, which is then rectified and mirrored to M_{inj} through M_2 - M_3 . V_{fg} and V_s stay constant while injecting due to the feedback structure provided by M_1 - I_2 . This process continues until V_{cg} gets close to V_{targ} , and the output current of the OTA decreases, which decreases the bias current of M_{inj} . As in [6], this reduction in the current through M_{inj} reduces the programming rate as the target is reached, thereby resulting in a better tradeoff between programming speed and accuracy. At this point, M_1 goes into the deep triode region, V_{cg} jumps up to the supply voltage, and injection is completely shut off.

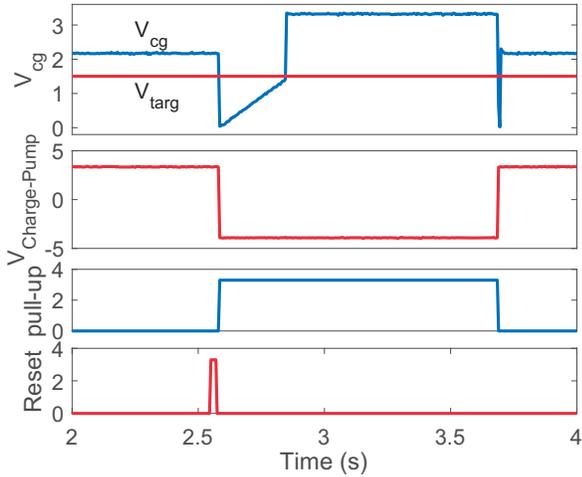


Fig. 6. Timing diagram of injecting an FG transistor to a desired value ($V_{targ} = 1.5V$), showing the control gate voltage, the output of the charge pump, pull-up voltage, and Reset voltage.

After being programmed up to a desired target, the circuit can be placed in a “read mode;” the charge pump is disabled which sends $V_{negative}$ to V_{DD} , and the two FG transistors are completely disconnected from the OTA and the other transistors of Fig. 5. A constant voltage is placed on V_{cg} , and “select” connects the source of $M_{circuit}$ to V_{DD} . The readout current, $I_{circuit}$, is connected to whatever circuit it is biasing, and it provides a constant and precise current source. We also use the same structure as read mode for the tunneling erasure process, except V_{tun} is elevated to approximately 15.5V and V_{cg} is set to ground.

When first placing a selected FG cell into program mode, V_{cg} frequently settles to a high voltage due to pre-existing biasing conditions from a previous read-mode setting or from being a non-selected device. Consequently, injection will never start because $V_{cg} > V_{targ}$, thereby shutting off the current in M_2 - M_3 . Transistors M_{Reset} and M_{bias} have been added to provide a short pulse to “reset” the feedback loop, and cause V_{cg} to go to its other stable point, which is $< V_{targ}$. M_{bias} is optional and can be used to limit the current during this short pulse.

Figure 6 shows a timing diagram of the programming process. Before starting injection, the memory cell is in read mode, so V_{cg} is at a fixed voltage, and the FG has been tunneled. When the injection process starts, V_{cg} starts from ground and ramps up quickly. When injection is finished, the charge pump is disabled and the memory cell is configured in read mode, again. The charge pump can remain on after the injection process is finished. This will not cause any further injection after V_{cg} has tripped to V_{DD} .

Figure 7 shows the performance of the memory cell and programmer combination. We programmed the memory cell to 21 different target values (V_{targ}) spaced evenly from 1.2V to 3.2V, and then we read the current through the $M_{circuit}$ in read mode. This process was repeated 100 times for each target voltage to check the repeatability of the programming process. Figure 7 (top) shows the average measured currents in read mode for different target voltages. Figure 7 (bottom) shows that the proposed structure can program currents as low as 1nA with a standard deviation $\leq 1.3\%$.

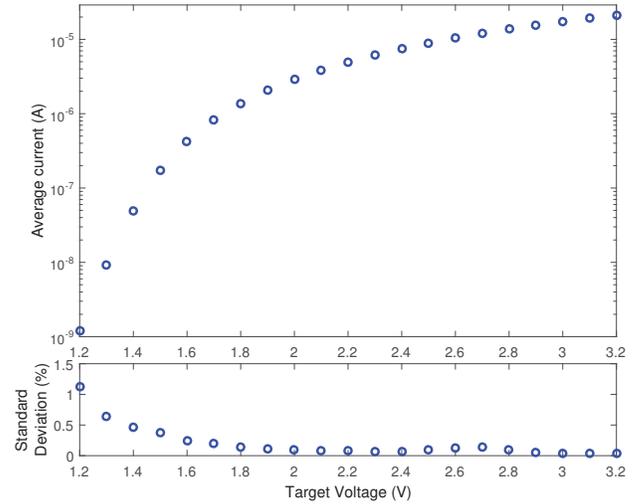


Fig. 7. (top) Average programmed current through $M_{circuit}$ in read mode over 100 programming iterations, for multiple V_{targ} values. (bottom) Standard deviation of the programmed current.

The goal of designing this memory cell is to use it in a large array of FG transistors. Therefore, we need to ensure that we can select a specific FG transistor when injecting and disable the rest of the FG transistors. To disable a specific FG transistor, we set “disable” such that the source terminal of the unselected $M_{in,j}$ transistors connects to ground (see Fig. 2). Then, even when the charge pump is enabled, the V_{sd} of the disabled transistors will be small and will not cause unwanted injection in the disabled memory cell.

VI. CONCLUSION

We have presented a compact floating-gate memory cell. Indirect programming of the FG transistors made it possible to use negative voltages to program the FG transistor and circumvent the issue of using selection circuits operating below ground. A closed-loop regulating charge pump was also presented to generate the negative voltages required for programming.

REFERENCES

- [1] P. Pavan, R. Bez, P. Olivo, and E. Zanoni, “Flash memory cells—An overview,” *Proceedings of the IEEE*, vol. 85, no. 8, pp. 1248–1271, 1997.
- [2] B. Rumberg and D. W. Graham, “A low-power and high-precision programmable analog filter bank,” *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 59, no. 4, pp. 234–238, April 2012.
- [3] C. Huang and S. Chakrabartty, “An asynchronous analog self-powered CMOS sensor-data-logger with a 13.56 MHz RF programming interface,” *IEEE J. Solid-State Circuits*, vol. 47, pp. 476–489, Feb. 2012.
- [4] P. Hasler and J. Dugger, “Correlation learning rule in floating-gate pFET synapses,” *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 48, no. 1, pp. 65–73, Jan. 2001.
- [5] P. D. Smith, M. Kucic, and P. Hasler, “Accurate programming of analog floating-gate arrays,” in *Proceedings of the IEEE International Symposium on Circuits and Systems*, vol. 5, 2002, pp. V-489–V-492.
- [6] B. Rumberg and D. W. Graham, “A floating-gate memory cell for continuous-time programming,” in *Proceedings of the Midwest Symposium on Circuits and Systems*, Aug. 2012, pp. 214–217.
- [7] B. Rumberg, D. W. Graham, and M. M. Navidi, “A regulated charge pump for tunneling floating-gate transistors,” *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 64, no. 3, pp. 516–527, March 2017.