An Analog Dynamic Memory Array for Neuromorphic Hardware

Matthias Hock, Andreas Hartel, Johannes Schemmel, Karlheinz Meier
Kirchhoff Institute for Physics
Ruprecht-Karls-Universität Heidelberg, Germany
Email: {mhock,ahartel}@kip.uni-heidelberg.de

***Abstract*—We describe an array of capacitor based cells capable of storing analog voltages and currents for highly configurable large-scale neuromorphic hardware. A novel refresh scheme based on content-addressable memory as well as a slow and simple voltage ramp generator is presented. The circuits have been simulated in a 65nm mixed-signal low power process. Key characteristics are an area consumption of 175 $\mu\mathrm{m}^2$ and a power consumption of less than 125 nW per stored value. A prototype chip has been designed and submitted for fabrication.**

I. Introduction

In VLSI implementations of neuromorphic hardware a large number of individually programmable parameters is desirable. This increases flexibility and broadens the range of networks that can be mapped to the hardware. Especially in deep submicron processes the possibility to compensate for device variation by calibrating individual circuits is important. The hardware developed within the BrainScaleS project is an example for a system with a large number of analog parameters [1]. In this system, about 20 parameters are assigned to each neuron, all of which can be configured independently. For an entire wafer-scale system containing 200 k neurons this leads to a total number of 4 M analog memory cells. This number shows that area and power efficiency of the analog memory cells are critical for such large-scale systems.

A common approach, also used in the system mentioned above, is the implementation of analog memory based on floating gates. As these devices provide very long storage times, typically longer than the duration of any experiment, no refresh cycles are required and it is not necessary to store the digital representation of the parameters on the chip. The downside of floating gates is the need for at least one additional and unusually high supply voltage, typically in the range of 8 to 30 V, which increases overall system complexity. Furthermore these devices require a rather complicated incremental programming process using feedback loops [2], [3]. Capacitive storage cells can be integrated and programmed easily but their storage time is limited. The cells need periodic refreshing during experiments and the digital representations of the values have to be stored on the chip. For experiments which involve learning processes that trigger changes of analog parameters or setups which allow for interactive parameter modification, the output of the cells should be stable during reprogramming. This is typically not the case for floating gate based memory.

II. Programming Scheme

This paper describes an architecture for an array of capacitive analog memory cells which store voltages and currents. To achieve high area and power efficiency a novel concept for the refresh process has been developed. A common method to update capacitive dynamic memory is to use a single DAC that programs all cells sequentially. For a large number of memory cells this requires a fast DAC and fast control logic reading the digital values from an SRAM block. In large arrays the minimum time required to update a single cell may not only be limited by the sample rate of the DAC but also by the RC delay of the wire which connects the storage cell and the DAC. The programming and refreshing scheme proposed here operates in a parallel fashion, using a single, slow ramp generator instead of a DAC. Every cell contains 10 bits of SRAM to store the digital representation of its target value locally. In the ramp generator a small constant current is continuously integrated on a capacitor, which leads to a slow and linear increase of the output voltage. This voltage slope is buffered and distributed across the whole array to every voltage cell, cf. Figure 1. Simultaneously a 10 bit digital counter is running which measures the time from the last reset of the capacitor C1. The counter value is also distributed to every storage cell, where asynchronous transmission gate logic is used to compare the value stored in its local SRAM to the current counter value. This concept can be compared to content-addressable memory which is used in digital high-speed search applications [4]. As soon as a match between external counter and internally stored value is detected, the cell updates its storage capacitor to the present value of the reference voltage V_{ref}. To prevent glitches in the asynchronous comparison logic due to the unmatched propagation delays of the counter signals, a counter based on Gray code [5] is used. Once the counter reaches its maximum value, the capacitor in the ramp generator is reset to ground and after a programmable delay the process restarts. This allows for all cells to be updated within one period of the 10 bit counter. Figure 2 shows a block diagram summarizing the architecture of the programming system, the details regarding the current cells will be discussed in section II-B.

A. Voltage Cells

In its simplest form a dynamic analog memory for voltages can be built from a single switch and a capacitor. To achieve reasonable storage times and to isolate noise caused by the

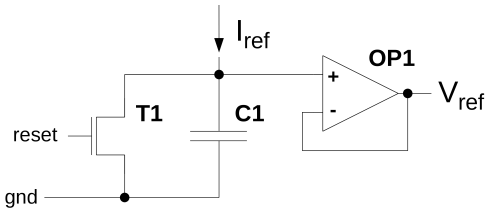

Fig. 1. Schematic of the circuit that generates V_{ref}. The constant current I_{ref} is integrated on the capacitor C1. The voltage on the capacitor is buffered by OP1 and distributed to the cells in the array.

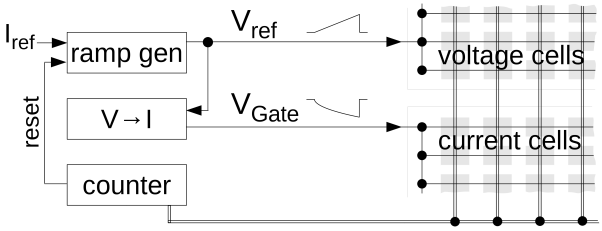

Fig. 2. Overview of the programming system.

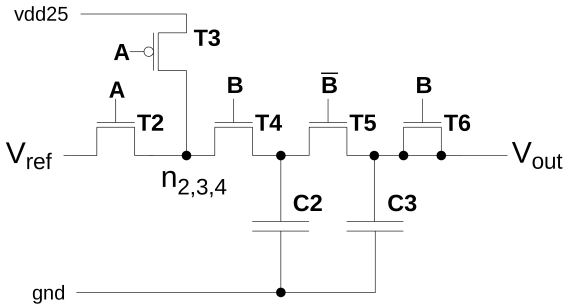

Fig. 3. Schematic of the analog part of a voltage cell.

programming process from the output, a more complicated architecture is necessary. Figure 3 shows the schematic of the analog part of a voltage cell. In total 5 transistors and 2 capacitors are used. Two signals, termed A and B in the following, are generated in the digital part of every cell to control the switches. To achieve long storage times it is necessary to minimize the leakage currents which discharge the storage capacitors C2 and C3. These need to be implemented as the gates of NMOS transistors, because the available MIM capacitors obstruct metal layers which are required for routing in the final system.

Tunneling leads to a significant leakage current through the gate of standard transistors in the 65nm process. Therefore it is necessary to build the analog part of the storage cell entirely from 2.5V thick-oxide transistors which are also available. Since the SRAM and the comparison logic is implemented using 1.2V standard transistors, two level shifting circuits generate 2.5V levels for signals A and B to control these transistors. The reverse current through the drain-bulk and source-bulk diodes is another mechanism which changes the amount of charge on the capacitors. This effect can be minimized by placing the switches in isolated substrate wells which are driven to exactly the same potentials as the stored voltages, as demonstrated in [6]. However, isolated wells consume a significant amount of additional area, therefore this is not an option for an array aiming at a high integration density. The bulk contacts of all NMOS (PMOS) transistors shown here are connected to ground (vdd25). A further mechanism limiting the storage time is the sub-threshold current through the switch T4. As long as the value in the cell is supposed to be constant, the signals A and B are both low. The node shared by T2, T3 and T4, denoted by $n_{2,3,4}$, is disconnected from the global reference voltage V_{ref} by T2 and pulled to vdd25 by T3. If in addition the voltage stored on C2 is larger than 200 mV, the gate source voltage of T4 is sufficiently negative to reduce the sub-threshold leakage current significantly.

The cell is programmed with positive pulses of A and B. The timing is such that the longer pulse of A embraces a shorter pulse of B, cf. Figure 4. The time difference between A turning high and B turning back low always corresponds to the time the counter holds one code. The pulse width of B can be configured. Signal A turning high stops node $n_{2,3,4}$ from being pulled to vdd25 and connects it to the reference voltage. The voltage at node $n_{2,3,4}$ has settled when B turns high, connecting C2 to the reference voltage and updating it to the new voltage. At the same time C3 is disconnected from C2 as T5 is controlled by the signal $\overline{B}$. As long as B is high, the output is still stable and undisturbed by the programming cycle. When B turns back low T4 becomes isolating, disconnecting the cell from the reference voltage. Simultaneously C2 is connected to C3 over T5. This updates the output voltage towards the new target value. The purpose of T6 is to counteract the coupling of signal B through the parasitic gate-source capacitance of T5 onto the output voltage. Due to the sequential connection of the capacitors C2 and C3 the output voltage is not updated immediately to a new voltage but exponentially approaches the target value. Depending on the capacitive load on the output, up to 15 programming cycles may be necessary until maximum accuracy is reached. Figure 5 shows the output of two voltage cells which are programmed to different values. As all cells in the array approach their target value, the amount of current drawn from the reference voltages approaches zero. Therefore the output resistance of the buffer and ohmic resistance of the wires distributing V_{ref} are not critical for accuracy. Figure 6 shows the deviation between the output of a voltage cell and a linear fit, which is below 5 mV over the dynamic range of the cell. Simulations have shown that the effect of process variations on the voltage cell is negligible, as long as the reference current of the ramp generator is calibrated. The impact of device mismatch has been investigated by Monte Carlo simulations of a voltage cell programmed to the digital values 16, 512 and 1008. The resulting standard deviation is below 6 μV in all three cases, which is significantly smaller than 1 LSB (1 LSB $\hat{=} 2\,V/1024$).

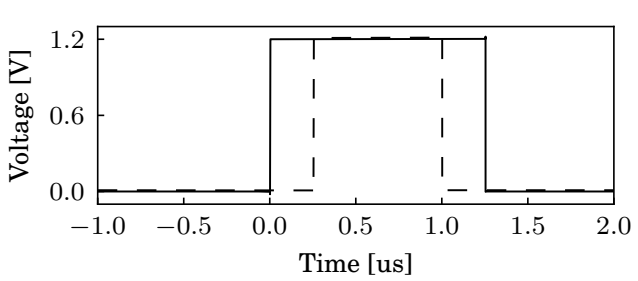

Fig. 4. Timing of the digital signals A (solid) and B (dashed) which control the switches in the cell.

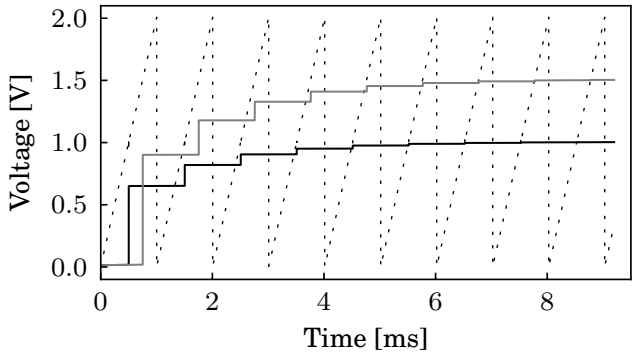

Fig. 5. Output voltage of two cells, programmed to the digital codes 511 (black) and 767 (grey) and reference voltage V_{ref} (dotted).

B. Current Cells

The structure of the current cells is very similar to the structure of the voltage cells, the schematic is shown in Figure 7. The additional PMOS transistor T7' converts the stored voltage into the output current I_{out}. Since voltages close to the supply voltage have to be stored to realize output currents in the order of a few nanoamperes, PMOS transistors are used as switches. Accordingly, the polarity of the control signals is inverted compared to the voltage cells. Using the linearly increasing V_{ref} directly to program the voltage on the storage capacitors would lead to a non-linear relation between

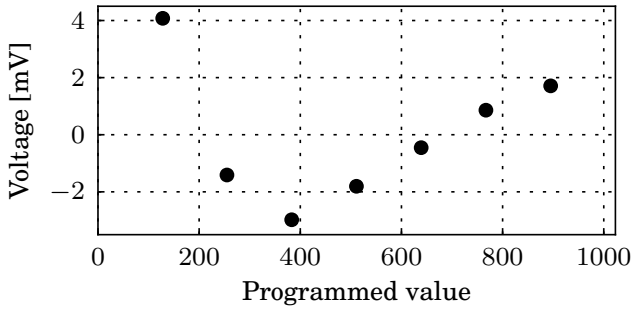

Fig. 6. Deviation of output voltage from a linear fit over the corresponding digital values.

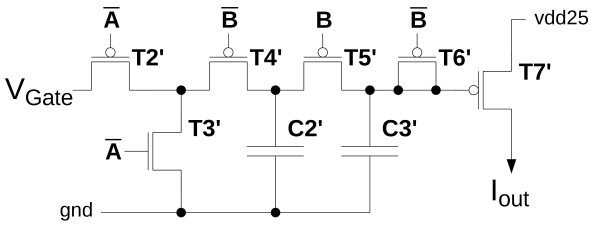

Fig. 7. Schematic of the analog part of a current cell.

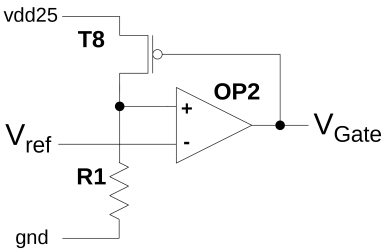

Fig. 8. Schematic of the circuit converting V_{ref} to V_{Gate} which is distributed to the current cells in the array.

digital codes and output currents. Furthermore this would cause a strong dependence of the output currents on global variations of the PMOS transistor characteristics. To avoid these problems, V_{ref} is used to generate a linearly increasing current using the operational amplifier OP2, resistor R1 and transistor T8, cf. Figure 8. The gate voltage of T8, denoted by V_{Gate}, is distributed over the array to every current cell. During a programming process T8 and T7 resemble a current mirror, compensating for most of the impact of global process variation. Figure 9 shows the output current during a single programming process, including the traces of simulations in slow and in fast corner. The remaining deviation from the typical case is in the range of 3 LSB (1 LSB $\hat{=} 2\,\mu\mathrm{A}/1024$) in either direction. The impact of device mismatch has been investigated by Monte Carlo simulations of a current cell programmed to the digital values 16, 512 and 1008. The resulting standard deviation is below 3 LSB in all three cases. Measurements conducted with a preceding prototype chip have shown that the variation predicted by the Monte Carlo models corresponds well to the mismatch observed in arrays of current mirrors, even if the transistors are located several hundred micrometers apart.

III. Implementation Details

An array of the analog memory cells described above has been designed and simulated. It features voltage cells providing a usable output range of 200 mV to 2 V and current cells providing an output range of 0 to 2 μA. Current and voltage memory cells are equal in size and cover 175 $\mu\mathrm{m}^2$ of chip area each. The transistors used as storage capacitors cover 40 $\mu\mathrm{m}^2$, corresponding to 22% of the total cell area.

The layout of the cells is structured so that up to 24 cells can be edge-connected to form a column. The ratio of voltage to current cells within one column can be chosen arbitrarily. All

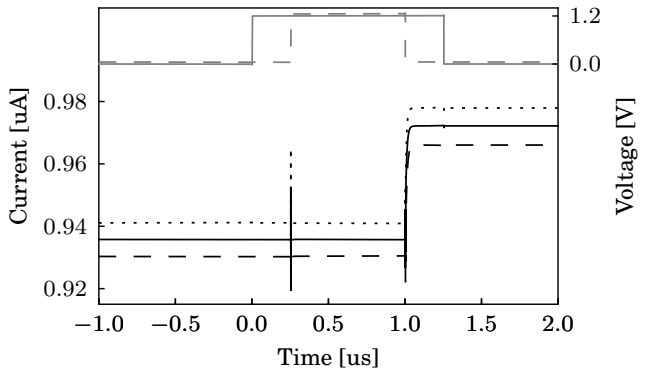

Fig. 9. Output of a current cell during a single programming process (left y-axis). The output current is shown for the corners typical (solid), fast (dotted), and slow (dashed). In grey the digital control signals A (solid) and B (dashed) are also shown (right y-axis).

output signals within one column are routed to the upper edge of the array. This allows for edge connection of one neuron circuit to one column on future chips.

The number of columns can be scaled without any changes to the circuits. A width of 128 columns has been successfully simulated. The supplementary circuits outside the array which generate V_{ref} and V_{Gate} are required only once, irrespective of the array size. Therefore their area consumption of $3200\,\mu\mathrm{m}^2$ is not critical in large-scale applications. The counter operates at a frequency of approximately 1 MHz, leading to a period of 1 ms for one programming cycle. For a prototype chip that has recently been submitted for fabrication, we chose an array size of 32 columns with 12 current and 12 voltage cells per column. According to measurements conducted with a preceding chip containing current cells with an architecture very similar to the ones described here, we expect the drift of the output current to be below $0.1\,\mu$A/s. In the present system this translates to a drift of less than 1 LSB per 20 ms. The total power consumption of the 24×32 cell array, operating at a refresh frequency of 1 kHz, is 0.18 mW. The contribution of every additional memory column to the total power consumption is $3.0\,\mu$W. For the rather small array discussed here, the supplementary circuits are responsible for about 50% of the total power consumption. Considering that the aspect of the supplementary circuits has not been optimized so far and that area consumption is not critical, significant reduction of their current consumption should be possible. If the storage time of the cells allows for a reduction of the refresh cycle frequency, as expected, the power consumption will be reduced almost by the same factor.

IV. Conclusion

An array of analog memory cells which meets the requirements for highly configurable large-scale neuromorphic hardware has been described. The integrated programming and refreshing scheme requires only a limited number of components and is rather power efficient.

All signals involved operate at frequencies at or below 1 MHz. The slow operation of the programming circuits is advantageous in terms of power consumption. As a disadvantage reprogramming of a parameter takes up to several milliseconds. When using highly accelerated hardware, as in the BrainScaleS project, this might have a significant impact on the total duration of a single experiment. One approach to improve performance and power efficiency is to dynamically adjust the timing of the programming process. When writing new values to the array, consecutive programming cycles ensure the fastest possible update of the outputs. If necessary the operating frequency of the counter can even be increased to achieve a higher update rate. When holding constant values, the storage time of the cells will allow to add delays between the refresh cycles to save power. With a fixed counter period of 1 ms the power consumed per parameter in a large array is 125 nW. The results of Monte Carlo and process corner simulations suggest that calibration of single cells is required for full 10 bit resolution, especially for the current cells. However, we expect one global calibration per chip, covering all global effects like process variation, variation of the resistor R1 or an offset voltage of the buffer OP1, to be sufficient to reliably achieve an effective precision of 8 bit.

The numbers given here are results of simulations using BSIM 4.5 models in the Cadence Spectre simulator. Leakage currents and charge injection in the MOS switches are difficult to simulate precisely. However these effects have significant influence on the performance of the cells. Therefore the numbers presented here have to be verified in silicon. A prototype chip containing the array as described has been submitted and will allow for a thorough characterization of the cells.

Acknowledgment

The authors wish to thank Simon Friedmann for helpful discussions about content-addressable memory. The work reported was funded by the Seventh Framework Program of the EC under grant agreement no. 269921 (BrainScaleS).

References

[1] J. Schemmel, D. Brüderle, A. Grübl, M. Hock, K. Meier, and S. Millner, "A wafer-scale neuromorphic hardware system for large-scale neural modeling," in *Proceedings of the 2010 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2010, pp. 1947–1950.

[2] D. Graham, E. Farquhar, B. Degnan, C. Gordon, and P. Hasler, "Indirect programming of floating-gate transistors," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 54, no. 5, pp. 951–963, 2007.

[3] Y.-D. Wu, K.-C. Cheng, C.-C. Lu, and H. Chen, "Embedded analog nonvolatile memory with bidirectional and linear programmability," *Circuits and Systems II: Express Briefs, IEEE Transactions on*, vol. 59, no. 2, pp. 88–92, 2012.

[4] A. Krikelis and C. Weems, "Associative processing and processors," *Computer*, vol. 27, no. 11, pp. 12–17, 1994.

[5] C. Savage, "A survey of combinatorial gray codes," *SIAM Review*, vol. 39, pp. 605–629, 1996.

[6] R. Wojtyna, "A concept of current-mode long-term analog memory for neural-network learning on silicon," in *Signal Processing Algorithms, Architectures, Arrangements, and Applications (SPA), 2008*, 2008, pp. 121–126.