Hybrid Discrete-Continuous Signal Processing: Employing Field-Programmable Analog Components for Energy-Sparing Computation

Employing Analog VLSI to Design Energy-Sparing Systems

Steven Pyle
Electrical Engineering
University of Central Florida
Orlando, FL
StevenDPyle@gmail.com

With power increasingly becoming the dominant limiting factor for today's computing needs, the promising methods of Hybrid Discrete-Continuous Architecture (HDCA) is being researched to aid in the advancement of energy saving computing solutions. As an older computational model, analog processing is coming back to help solve today's computational needs at efficiencies unachievable with purely digital systems. Utilizing both analog and digital components, HDCA aims to combine the two domains to do what each domain does best, together. With computational energy-efficiency approaching 10,000 times the computational efficiency of purely digital systems, HDCA is poised to be a major breakthrough in the coming years. In this paper we will discuss what HDCA is, why we are researching it, the applications HDCA can be used for, and various experiments and simulations demonstrating the energy saving capabilities of HDCA.

Keywords—Analog, continuous, hybrid, energy-efficient, processor architectures, Field programmable analog arrays, FPAAs, RASP, HDCA

I. Introduction

With technology trends increasingly becoming limited by power and transistors nearing atomic limits in size, it is becoming increasingly difficult to continue Moore's Law, and therefore research interest is considering exploration of significantly different computing methods than those in use today. HDCA aims to utilize the device physics of transistors to perform energy efficient computations while integrating digital circuits to allow for reliability and repeatability, which has always been a primary concern with analog computers. This paper is organized with first introducing the necessities of today's computing trends and why HDCA will help develop solutions for those needs, then we will introduce various applications of which HDCA techniques can be applied, we will analyze various experiments and simulations utilizing HDCA and their results, and finally, we will assess the approaches introduced in this paper and identify future exploration that could grow the field.

II. Motivation for Hybrid Discrete-Continuous Architecture

A. Analog vs Digital

When processing information is received via real-world phenomenon, a major inefficiency arises when one uses strictly digital methods [1]. Processing such information digitally usually ends up using numerical algorithms, which can take many steps, adding to the inefficiency [1]. Analog approaches, on the other hand, have the distinct advantage of being able to utilize the device physics of transistors to implement a certain computation in a single analog block, allowing for performance and efficiency improvements [1].

Despite the benefits of analog computations, they are not without their limitations. Analog processing is much more error-prone than its digital counterpart, as well as much more difficult to reproduce [1]. Device mismatches and thermal noise have a multiplicative effect through the various computational stages in an analog system, leading to losses in precision. However, by combining the repeatability and accuracy of the digital domain with the efficiency and performance of analog systems, we can find a compromise between the two allowing us to develop both reproducible and energy-efficient computational solutions [1].

B. Energy Savings

Because technology is becoming increasingly mobile and supercomputing systems require massive server farms, energy, not raw computing power, has become the dominant limiting factor for today's computing needs. Because of this, research is continually being developed to reduce power consumption in computational systems in order to improve battery life of mobile devices and decrease the massive power and cooling needs for supercomputing systems. HDCA is primed to contribute immensely to the reduction of power requirements in computing systems. HDCA, as we will show later in this paper, can theoretically provide upwards of 10,000 times the computational efficiency of certain computations than purely digital implementations. This equates to a 20 year leap ahead of Gene's Law for DSP ICs as shown in figure 1. To put this into perspective, the energy efficiency increase possible through HDCA would be greater than the advances in DSP chips from the first marketed DSP chip to the ICs of 2005 [2].

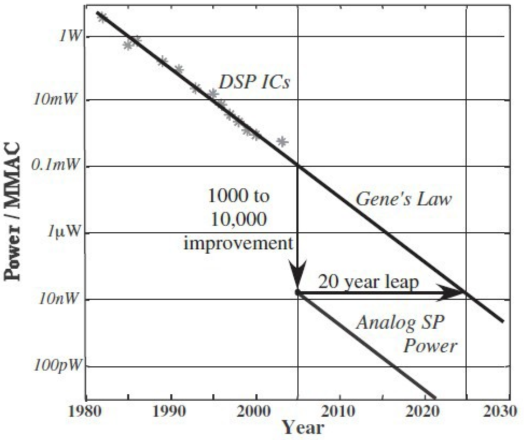


Fig. 1. HDCA can theoretically provide a 20 year leap ahead of today's ICs on the Gene's Law curve [2].

C. Performance

Along with the energy-efficient computations that HDCA can provide, there are also performance benefits for utilizing HDCA. Purely digital systems depend heavily on the switching characteristics of the transistors making up the various gates, and setup and hold times also must be taken into account in order to make sure the system runs reliably without latching improper values into flip-flops. Analog systems, however, rely solely on the settling time of the module, and because of the highly parallel nature of analog systems, increasing the computational load of the module has little effect on the settling time, and therefore the overall performance of the module [3].

One thing to note, however, is the typical inaccuracies of analog systems. Since analog systems are affected significantly more than digital systems by device mismatches, component inaccuracies, and various transient noises, the level of accuracy obtainable from analog circuits is generally quite lower than its digital counterpart [1]. What's interesting to note though is that in certain applications, HDCA can use analog circuits to approximate good solutions and digital circuits to “refine” the solution generated by the analog circuits to a desired accuracy, which still takes less energy than performing the entire calculation by digital systems alone [1].

III. Applications

Currently, HDCA is only applicable to certain computational applications generally in which one may perform the analog computation as close to the real-world continuous input as possible, and then using ADCs to relay the processed information to digital systems for further computation, storage, or output [2]. These applications typically involve signal processing, ordinary differential equations, learning algorithms, and seeding for high-accuracy iterative digital algorithms [1].

A. Signal Processing

Many applications where we can use HDCA to improve energy-efficiency and performance are situations where we would normally use a DSP to perform the entirety of the process. With HDCA, we can significantly reduce the performance burden of both the DSP and the ADC by implementing some analog circuits, usually with a reconfigurable programmable analog system chip, to perform some computation on the original continuous signal before handing it off to an ADC for conversion for the DSP [2]. This breakdown is shown in figure 2.

Typically, the computation is broken down in such a way that it can be implemented by vector-matrix-multiplication (VMM), which is quite an easy circuit to implement in the continuous domain. VMMs are readily available for many different transforms such as discrete-cosine transform (DCT) and discrete-fourier transform (DFT), and as such, can be easily implemented in the continuous domain as will be shown later in the paper [2] [3].

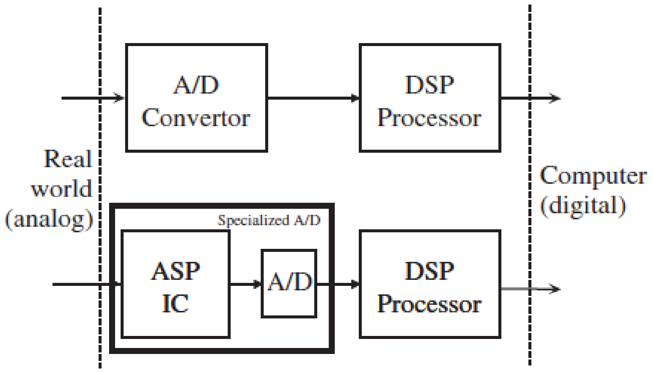


Fig. 2. Decomposition of typical purely digital signal processing into HDCA processing [2].

B. Learning Algorithms

A significant amount of work has already been done to apply analog systems to neural networks [1]. Analog systems have already been shown to improve the power-efficiency and performance of branch prediction by using a neural predictor [4]. Because of the device physics of transistors, analog components are a good match for artificial neural networks since one may use the transistors to compute the weights of the neurons [1].

C. High Accuracy Applications

Even though computations in the continuous domain are typically inaccurate compared to their discrete counterparts, HDCA can be utilized to improve the power-efficiency and performance of high accuracy applications by accelerating seeding and iterative steps [1].

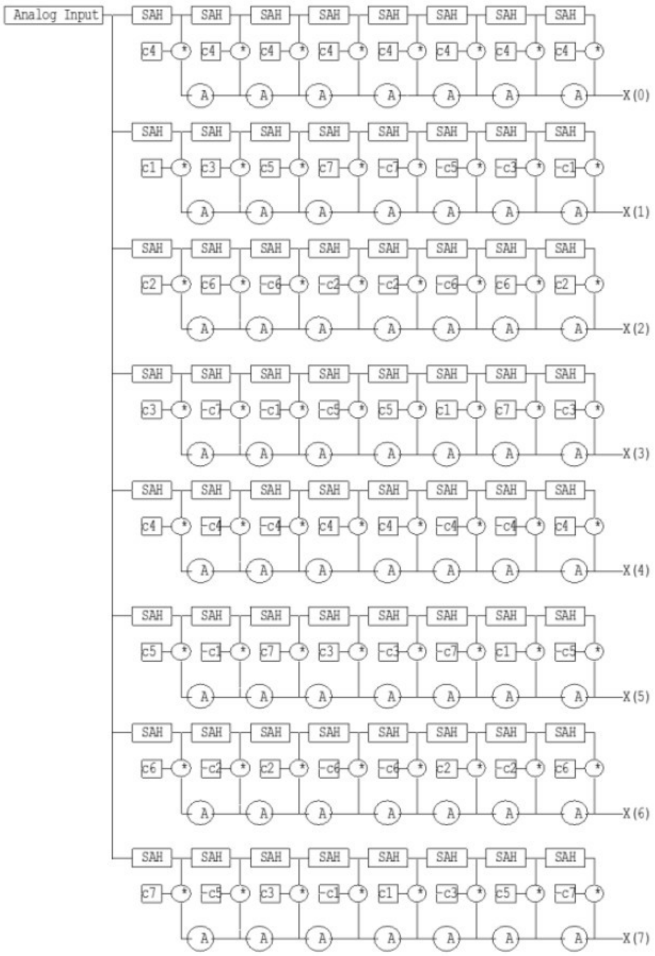

Analog Input
SAH
c4 c4 c4 c4 c4 c4 c4 c4
c1 c3 c5 c7 -c7 -c5 -c3 -c1
c2 c6 -c6 -c2 -c2 -c6 c6 c2
c3 -c7 -c1 -c5 c5 c1 c7 -c3
c4 -c4 -c4 c4 c4 -c4 -c4 c4
c5 -c1 c7 c3 -c3 -c7 c1 -c5
c6 -c2 c2 -c6 -c6 c2 -c2 c6
c7 -c5 c3 -c1 c1 -c3 c5 -c7
*
A
X(0)
X(1)
X(2)
X(3)
X(4)
X(5)
X(6)
X(7)

Fig. 3. Analog VLSI implementation of 8-point DCT

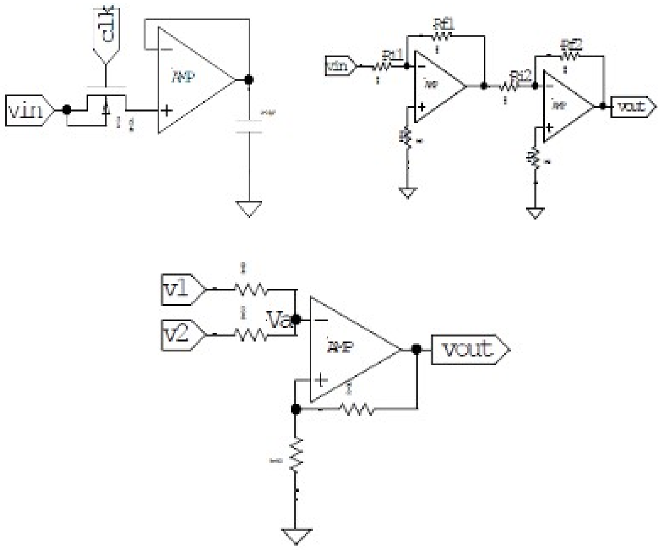


Fig. 4. Analog VLSI op-amp circuits for sample-and-hold (top-left), multiplication (top-right), and addition (bottom) [5].

Many computational algorithms utilize iterative methods in order to converge solutions. Convergence is typically determined by how close the initial values (seeds) are to the final solution, and how much work it takes to compute each successive step [1]. Many algorithms, especially for non-linear problems may not ever converge unless their initial values are sufficiently close to the true solution [1]. HDCA can improve these high-accuracy iterative algorithms by approximating the initial seed solutions to the algorithms, allowing the digital algorithm to take over and continue improving the result until the desired accuracy is met [1]. This reduces the amount of iterative steps needed and can help difficult to converge algorithms find a proper solution without testing many initial seeds [1].

HDCA can also help speed up high accuracy applications by “boosting” the iterative steps of the algorithm if the intermediate computations can be implemented in the continuous domain [1].

IV. Experiments and Results

For this next section, we will be going over the experiments and results from various papers where the author(s) utilized HDCA to implement real-world signal processing computations and achieved substantial energy savings.

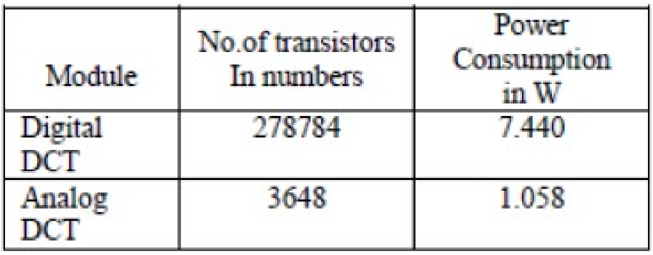
A. Analog VLSI Architecture for DCT

The goal of [5] is to introduce the concept of utilizing classic op-amp sample and hold, addition, and multiplication circuits shown in figure 4. With these principle circuits, the paper implements vector-matrix-multiplication to realized an 8-point DCT computational module by successive stages of multiply/accumulate operations. They did this by realizing the circuit in figure 3 to break down the continuous input signal into the first 8 DCT signals.

With this, they compared the transistor count of the digital implementation and the analog version shown by simulation in TSPICE. The transistor count and power consumption of the architectures are shown in figure 5. From these results, we notice that the analog DCT implementation on requires 1.3% of the transistor count of the digital counterpart, with a power reduction of 7 fold.

B. Low-Power Programmable Signal Processing

In [2], we present a reconfigurable analog signal processor (RASP) that was developed at Georgia tech and some of the applications that they suggest for it. The various computational elements included on the RASP are shown in Figure 6.



Module	No.of transistors In numbers	Power Consumption in W
Digital DCT	278784	7.440
Analog DCT	3648	1.058

Fig. 5. Analog and Digital implementation of DCT transistor count and power consumption [5].

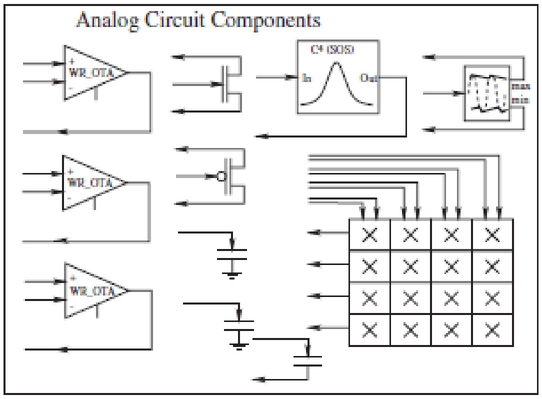


Fig. 6. RASP computational elements [2]

First, we notice that the RASP includes the components to realize analog VLSI from the previous paper, but also includes a switch matrix of floating-gate transistors, which can be used themselves as processing elements. With these floating-gate transistors, they were able to realize a similar vector-matrix-multiplier to the previous paper, but with much fewer elements and a much simpler structure shown in figure 7. With this, they were able to achieve a computational efficiency of 4 MMAC/uW, which when compared to the best DSCP IC's of 2005 with a computational efficiency of 4-10 MMAC/mW.

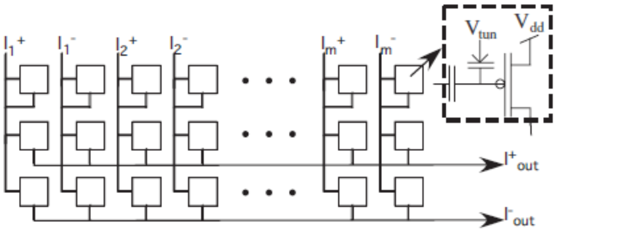


Fig. 7. Floating-gate transistor implementation of a Vector-matrix-multiplier [2].

This leads to a factor of 1,000 times greater computational efficiency than the digital implementation. [2] even suggests that analog computation could be up to 10,000 times more computationally efficient than their digital counterparts.

Hasler et al [2] then suggests that one example of using a VMM similar to this would be for direct computation on photo-sensors to for a DCT for JPEG compression or other compression algorithms. The architecture to realize this can be found in figure 9. [2] states that the transform imager would require roughly 1mW as a single chip solution; with this in comparison to the roughly >1W solution for the standard digital implementation, we can anticipate substantial savings by utilizing HDCA for the image processor.

Reference [2] also goes over utilizing the floating-gate transistors of the RASP to realize adaptive filters and neural networks. The RASP is designed to be dynamically reconfigurable, and therefore algorithms can be developed with it to dynamically alter the “weights” associated with the floating-gate transistors to implement the adaptive filters and neural networks. Figure 8 shows how these algorithms are implemented. Although there wasn't any raw data provided, [2] mentions that a 128x128 synapse array can be realized using this topology to operate at under 1mW; purely digital implementations consume 3-10W for the same number of synapses. This leads to a efficiency gain of up to 10,000 fold.

These implementations of the RASP are in addition to other experiments performed on the platform, such as a noise suppression algorithm for speech recognition [7] and a low-power reprogrammable analog classifier [8].

V. Approach assessment

In this section, we assess the benefits and drawbacks of the two approaches discussed in section IV. We are interested in how these HDCA methodologies energy-efficiency repeatability, precision, programmability, and future development perform.

Utilizing analog VLSI such as in [5], they implement the DCT by directly using well known standard analog signal processing circuits with op-amps, resistors, capacitors, and MOSFETs. They showed a ~7 fold decrease in energy consumed compared to the purely digital implementation, but this begs to wonder how much better the efficiency could be if one were to use analog solutions which did not depend on resistors, as resistors are inherently very inefficient. The repeatability of the system if it were to be mass-produced, reconfigurable, etc. hinges on how much device mismatch is present as well as thermal noise introduced into the system. Inaccuracies in device properties can cause a domino effect of degraded signal quality which can become quite large, especially as the systems grow larger and larger. Such a system is not inherently programmable, since the circuits are made of individual ready-made devices such as resistors and capacitors; one could, however, introduce programmability by utilizing floating-gate transistors as the programmable resistors in the circuit, or to have banks of devices which can be routed as desired. Overall, this method of analog computation seems quite archaic, and that other more creative solutions can be developed to provide better efficiency, repeatability, precision, and programmability. Since other solutions can provide much better qualities for signal processing applications, we deem the applicability of this platform for future development to be rather low.

The second analog signal processing approach we explored in [2] improves on all the qualities we are looking for in an HDCA application. The energy-efficiency of such an application has been shown to approach up to 10,000 times the efficiency of it's digital counterpart; much more than the 7 times shown by the previous approach. This has to do with the creative way floating-gate transistors are used as programmable computational elements, eliminating a lot of the transistors needed for the op-amps in the previous approach, as well as not requiring resistors. The repeatability and precision of such a system is still rather reliant upon device mismatches and thermal noise, but its ease of programmability can allow for a myriad of adaptive algorithms to be implemented, which can alleviate much of the downfalls of analog systems.

Overall, the creative way that [2] implements signal computations in the analog domain really shows that there is very much still a lot out there that we have to discover in analog, digital, and hybrid computational systems. Reference

[2] shows that there really are some large efficiency savings to come from utilizing computations in the analog domain and that we should continue researching these techniques to further continue the improvements we've seen from Moore's law since its inception.

VI. Future work

In this section we are going to discuss two areas of future work in HDCA, which may continue and help drive the improvements in efficiency as well as programmability and repeatability that we've seen so far.

Much like how [2] suggests using the floating-gate transistor networks of the RASP to implement adaptive filters and neural networks, I believe that modifying such a system to adapt the weights of the transistors, not in accordance with an adaptive filter, but to the actual desired weights of the transistor. Such a system can dynamically adapt the weights of the transistor such that they are as accurately tuned as possible, given the precision of the A/D converter and the level of charge able to be adjusted on the floating-gate transistor. This system could be implemented by inserting some digital feedback, which takes the input/output data of a test signal and determines whether or not charge should be added/taken from the floating-gate. Reference figure 10 for a block diagram view of such an adaptive system. This can also help increase the programmability and repeatability of analog processing systems by simply programming what the weight values should be in the digital feedback. This would allow the system to automatically adjust the weights of the floating-gate transistors intrinsically.

Since we are just recently beginning to develop more creative solutions to the utilization of analog computations to improve our processing needs, I suggest that an interesting exploration of the hybrid analog-digital domain could be accomplished by the utilization of a genetic algorithm (GA). If one were to modularize the analog sub-systems such as adders, multipliers, and integrators as well as modularize digital subsystems and develop methods for these sub-systems to interchange and relay information between each other at a module level, then one would be able to implement a GA to explore the hybrid analog-digital domain. What could come out of this? We'll we aren't too sure, but just like Thompson in [6], a simple exploration of a new domain with a GA can yield very interesting and possibly field-changing or field-developing results. Such a system has not been implemented before, and the results could be revolutionary or uninteresting, but it should still be performed nonetheless.

VII. Conclusion

With the increasing need for low-power systems, getting as much computation for a fixed power budget becomes more and more critical. With analog computation taking some of the load traditionally left to the digital domain, significant power savings can be achieved. In this paper we observed substantial power savings when using analog VLSI to process a DCT, which can be used for compression algorithms, we also observed the RASP, with a bevy of analog computational elements, able to implement vector-matrix-multiplication and adaptive filters with energy savings of up to 10,000 times the digital implementation, and considered other utilization's of analog VLSI in tandem with digital VLSI to save energy, such as enhancing the "seeding" of high-accuracy iterative algorithms and "boosting" the intermediate computations of such algorithms. By continuing research into this relatively new field by researching new ways to dynamically adapt the analog parameters to increase the precision, repeatability, and programmability of analog computations as well as employ GAs to explore the hybrid analog-digital domain, we can further uncover new ways to save computational energy and continue driving the technological demands of today's industry and consumer electronics.

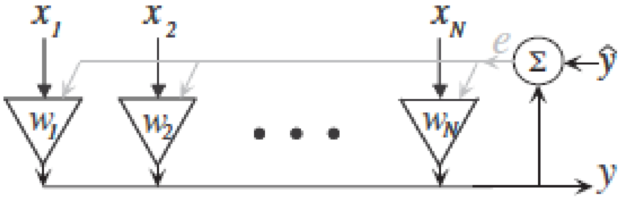


Fig. 8. Utilizing RASP's floating-gate transistors to realize adaptive filters and neural networks [2].

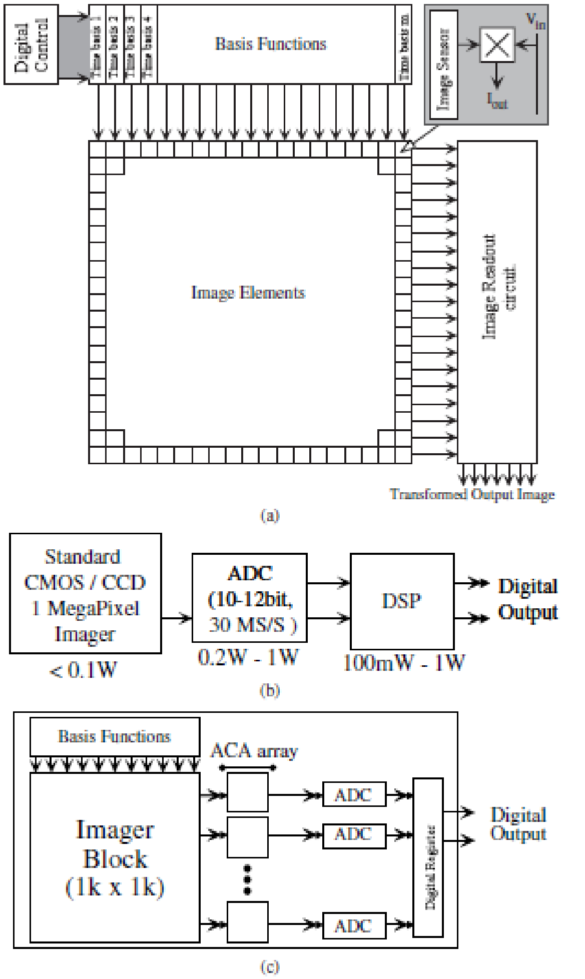


Fig. 9. Direct analog computation implemented onto a single imager chip. (a) Top view of the matrix imager. (b) Standard digital JPEG implementation. (c) HDCA single chip JPEG implementation [2].

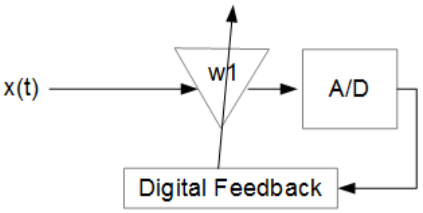


Fig. 10. Digital feedback for dynamic adaptation of floating-gate weights.

References

[1] S. Sethumadhaven, R. Roberts, Y. Tsividis, “A Case for Hybrid Discrete-Continuous Architectures,” IEEE Computer Architecture Letters, vol. 11, No. 1, January-June 2012

[2] Hasler, P., "Low-power programmable signal processing," *System-on-Chip for Real-Time Applications, 2005. Proceedings. Fifth International Workshop on* , vol., no., pp.413,418, 20-24 July 2005 doi: 10.1109/IWSOC.2005.83

[3] S. Suh, A. Basu, C. Schlottmann, P. Hasler, J. Barry, “Low-Power Discrete Fourier Transform for OFDM: A Programmable Analog Approach,” in IEEE Transactions on Circuits and Systems, vol. 1, No. 2, February 2011, pp. 290–298.

[4] R.S. Amant, D.A. Jim´enez, and D. Burger. Low-power, highperformance analog neural branch prediction. In *Proceedings of the 2008 41st IEEE/ACM International Symposium on Microarchitecture-Volume 00*, pages 447–458. IEEE Computer Society Washington, DC, USA, 2008.

[5] M. Thiruveni, M. Deivakani, “Design of analog vlsi architecture for dct,” International Journal of Engineering and Technology, vol. 2, no. 8, August, 2012.

[6] A. Thompson, “Silicon Evolution,” In *Proceedings of the First Annual Conference on Genetic Programming* (GECCO '96). . 1996. MIT Press, Cambridge, MA, USA, 444-452.

[7] Ramakrishnan, S.; Basu, A.; Leung Kin Chiu; Hasler, J.; Anderson, D.; Brink, S., "Speech Processing on a Reconfigurable Analog Platform," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on* , vol.22, no.2, pp.430,433, Feb. 2014 doi: 10.1109/TVLSI.2013.2241089

[8] Ramakrishnan, S.; Hasler, J., "Vector-Matrix Multiply and Winner-Take-All as an Analog Classifier," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on* , vol.22, no.2, pp.353,361, Feb. 2014 doi: 10.1109/TVLSI.2013.2245351